Improved Weak Gravitational Lensing Using Generative Adversarial Networks

Michel Dellepere
Department of Physics
Stanford University
mdelle1@stanford.edu

Amay Aggarwal
Department of Computer Science
Stanford University
amayagg@stanford.edu

Andrew Ying
Department of Computer Science
Stanford University
androo@stanford.edu

1 Background

Gravitational lensing occurs when massive astronomical objects create gravitational fields that distort the path of light through space. One particular sub-regime is known as Weak Gravitational Lensing (WGL), a phenomenon that physicists use as a tracer for dark matter. In cosmological modeling, WGL is considered one of the most powerful tools since it probes the geometry and growth structure of our Universe, and allows us to reconstruct distribution of matter along our line of sight. A common application of lensing has been in the matter power spectrum, which allows astrophysicists to make predictions about cosmic acceleration.

Different cosmological models give rise to dark matter distributions, which are simulations of cosmic structures visible (and invisible) to the naked eye. Astrophysicists use N-body techniques to represent these distributions as interacting bodies that behave according to the laws of gravity. A major drawback of N-body simulators, however, is that they are computationally expensive, requiring long periods of time before producing useful results. Deep Generative Models are promising candidates for fast, but precise, modeling of WGL. A generative model allows us to create virtually unlimited samples in a short period of time, generating data for astronomers to use in their research.

2 Related Work

Most prior work in weak gravitational lensing has been done by hand with N-body simulations. There are many types of N-body simulations, such as `DUSTGRAIN-pathfinder`, which extends general relativity principles for massive neutrinos, and the `Halo` model, which takes as input the halo population within a desired light-cone and the linear power spectrum of the underlined cosmological model [3, 2]. As mentioned above, these methods are lengthy, ranging from 750-1550 hours (1-2 months) depending on the grid dimensions of the generated image.

Because weak gravitational lensing using GANs is still a relatively nascent field, there are two keystone works in the area [7, 9]. Mustafa et al. demonstrated that a deep convolutional GAN (DCGAN) model produced weak lensing convergence maps that are described by, with high statistical confidence, the same summary statistics as the fully simulated maps [7]. Rodriguez et al. utilize the same model architecture (DCGAN), but train it on a different dataset generated by N-body simulations. They found qualitative and quantitative results that agree with Mustafa et al.: generated images appear indistinguishable from real images, and error rates for power spectrum were within 15%. To our knowledge, these are the main works in the field, and thus can be viewed as state-of-the-art.

Training GANs is notoriously difficult as they often suffer from lack of convergence, caused by unstable updates of the loss function towards the end of training [1]. Both of the works specified in the previous paragraph mention grappling with these challenges. Specifically, Mustafa et al. identified "the most useful performance improvement" to be training stabilization. To remedy this issue, researchers have proposed a *geometric* GAN (GeoGAN), which is a geometric generalization of a traditional adversarial network [6, 5]. Additionally, researchers have suggested that rather than estimating the probability that input data is real, the goal of a GAN's discriminator should instead be estimating the probability that real data is more realistic than randomly sampled fake data. This model has been termed the relativistic GAN [4].

3 Problem and Objectives

Due to timely and computational constraints, we are not able to train entirely on CosmoGAN dataset of 80,000 cosmological images. We also acknowledge that CosmoGAN paper achieved extremely high accuracy, due to a condition which terminated the algorithm when it reached a certain level of precision. This condition was in place because CosmoGAN suffered from unstable updates, especially near the end of training, reporting that "the generator varied uncontrollably".

Our goal is to train on a smaller data sample (16k images) and create a robust model, that is both continuously stable and highly accurate. Therefore, we advance this model as a *proof of concept* that could very well be used in larger datasets given adequate computational resources.

4 Dataset

Our dataset (private; obtained from [7]) contained 16,000 cosmological maps (256×256) generated via N-body simulations. Three of those maps are reproduced above for clarity.

While we did not do any preprocessing, these images were scaled down from 1024×1024 after creation to facilitate training. We followed the process detailed in [7] for creation of the validation set, sampling from the entire set of maps for creation of the validation set because GANs are generally not susceptible to memorizing the training set. Because our architecture was a convolutional neural network, features were the process of convolving over each batch of images.

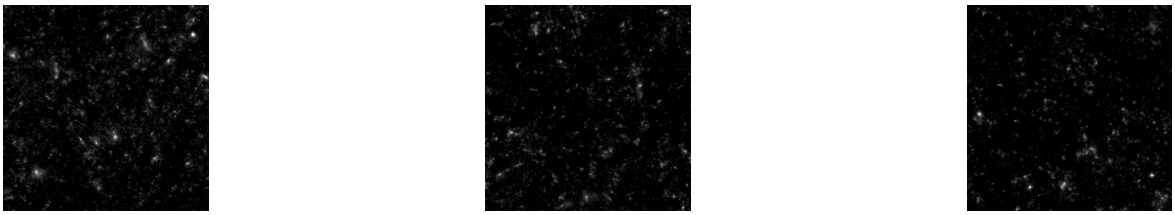


Figure 1: Example Training Data

5 Methods and Experiments

5.1 Geometric Adversarial Training

Original GANs solve the following minimax game:

$$\min_G \max_G L(D, G)$$

using a standard cross-entropy loss,

$$L(D, G) := \mathbb{E}_{x \sim P_s}[\log[D(x)] + \mathbb{E}_{z \sim P_z}[\log(1 - D(G(z))]$$

where P_s is the true sample distribution; P_z is the latent variable distribution; $D(x)$ is the discriminator output on input $x \in \chi$, and $G(z)$ is generator output on latent variable z. Training GANs is notoriously difficult as they suffer from lack of convergence, caused by unstable updates of the loss function towards the end of training [1]. To remedy this issue, Lim et. al. [2017] propose a geometric generalization of the GAN framework that uses a support vector machine (SVM) separating hyperplane, composed of the following geometric operations in feature space:

- Finding a separating hyperplane for a linear classifier.
- Discriminator parameter update away from from the separating hyperplane using gradient descent.
- Generator parameter update along the normal direction of the hyperplane

Given some feature mapping $\Phi : \chi \to \xi$, the primal form of the soft-margin SVM is given by the reformulated optimization problem

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} (\xi_i + \xi i')$$

subject to constraints

$$\langle w, \Phi(x^{(i)}) \rangle + b \geq 1 - \xi_i, i = 1, ..., n$$
$$\langle w, \Phi(x^{(i)}) \rangle + b \leq \xi_i' - 1, i = 1, ..., n$$
$$\xi_i, \xi_i' \geq 0, i = 1, ..., n$$

where C is a fine tuning parameter and ξ, ξ' are slack variables. More lucidly, the discriminator acts like a standard SVM in that it wants to maximize the distance of real and fake samples from hyperplane. The generator updates in the direction of the normal vector to the plane since it wants fake samples to be perceived as real, and is the added constraint in the optimization.

Lim and Ye [2017] show that in sample limit $n \to \infty$, a geometric GAN should mathematically converge to a Nash Equilibrium, where real and generated images can no longer be separated.

5.1.1 Loss Functions

Interpreting the gradient descent parameter updates in Lim and Ye [2017], we are effectively minimizing a hinge loss,

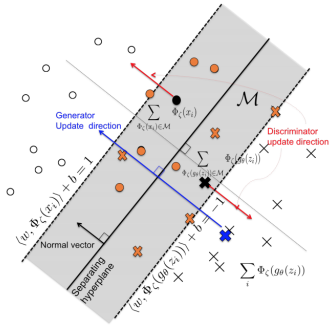
$$\mathcal{L}_D(\zeta, \theta) \coloneqq \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - D_\zeta(x^{(i)})) + \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 + D_\zeta(g_\theta(x^{(i)})))$$

$$\mathcal{L}_G(\zeta, \theta) \coloneqq -\frac{1}{n} \sum_{i=1}^{n} D_\zeta(x^{(i)})$$

where L_D and L_G are the discriminator and generator loss functions, and ζ and θ are the parameters of the respective deep networks.

5.1.2 Experiments

We train the GeoGAN-DCGAN hybrid model with the modifications discussed above. In addition, we experiment with a relativistic loss function by loading standard GeoGAN weights until epoch 85, and from there training with a relativistic discriminator (Jolicoeur et. al.). Relativism is an important concept since it makes parameter updates based the difference in discriminator scores of real and fake images. We observe very similar results between both models, but focus moreso on the evaluation of standard GeoGAN.



	Activ.	Output Shape	Params.
Latent	-	64	-
Dense 5×5	-	$512 \times 16 \times 16$	8.5M
VirtualBatchNorm	ReLU	$512 \times 16 \times 16$	1024
TConv 5×5	-	$256 \times 32 \times 32$	3.3M
VirtualBatchNorm	ReLU	$256 \times 32 \times 32$	512
TConv 5×5	-	$128 \times 64 \times 64$	819K
VirtualBatchNorm	ReLU	$128 \times 64 \times 64$	256
TConv 5×5	-	$64 \times 128 \times 128$	205K
VirtualBatchNorm	ReLU	$64 \times 128 \times 128$	128
Conv 5×5	Tanh	$1 \times 256 \times 256$	1601

	Activ.	Output Shape	Params.
Input map	-	$1 \times 256 \times 256$	-
Conv 5×5	-	$64 \times 128 \times 218$	1664
BatchNorm	LReLU	$128 \times 64 \times 64$	256
Conv 5×5	-	$256 \times 32 \times 32$	819K
BatchNorm	LReLU	$256 \times 32 \times 32$	512
Conv 5×5	-	$512 \times 16 \times 16$	3.3M
BatchNorm	LReLU	$512 \times 16 \times 16$	1024
Minibatch Discrimination	Linear	131328×1	$\sim 82K$
Conv 5×5	-	$513 \times 16 \times 16$	4.4M
BatchNorm	LReLU	$513 \times 16 \times 16$	1024
Linear	Linear	1	131K

Table 1: Above: SVM separating hyperplane and architectures for our model generator and discriminator

5.2 Deep Convolutional Architecture

We use a similar approach to the standard DCGAN model implementation [8], with modifications relevant to batch normalization and layer stack composition.

5.2.1 Generator Network

The input **z** to the generator is a 64-dimensional latent vector such that $z_k \sim \mathcal{N}(0, 1)$ for $0 \leq k \leq 64$. To learn proper correlations among all possible combinations of input features z_k, the first layer is fully-connected. Standard DCGAN implementations in astro-imaging reshape the output of the first layer into a stack of feature maps and use a Dense layer plus three layers of 5×5 transposed convolutions with two strides, activating using ReLU, except for the last layer which uses tanh. While batch normalization (BN) in intermediate output layers has significantly improved DCGAN performance (Santurkar et. al. [2019]), we instead use *virtual batch normalization (VBN)*, where samples $\{x^{(i)}\}_{i=1}^{i=m}$ are normalized based on the combined statistics of the batch and a reference batch fixed at the beginning of training. Outputs generated using regular BN in neural networks tend to be highly dependent on other inputs of the same batch (Goodfellow et. al. [2016]). Since our generated images should ideally be independent samples from the same distribution, we consider this a reasonable addition.

5.2.2 Discriminator Network

The first four layers of the discriminator are standard DCGAN implementation, composed of four convolutions activated with LeakyReLU ($\alpha = 0.2$) and batch normalization. We preemptively focus on adding layers to increase the stability of the generator. Rodriguez et. al. [2018] report that in inhomogenous image types, such as ones that contain dense regions with halos, the standard DCGAN training algorithm has a higher chance of focusing on a small subset of the distribution. To preemptively hinder what seems like mode collapse, we include a layer known as minibatch discrimination, a strategy developed by Goodfellow et. al. [2016] that looks at the closeness of images in a minibatch. Specifically, if $f(x^{(i)}) \in \mathbb{R}^n$ is an feature produced by some intermediate layer of the discriminator, we multiply $f(x^{(i)})$ by a trainable tensor $T \in \mathbb{R}^{m \times p}$ to get a matrix $M \in \mathbb{R}^{n \times m \times p}$. We compute the similarity kernel for every row in M as,

$$K_b(f(x^{(i)}), f(x^{(j)})) = \exp\left(-\left\|M_{i,b} - M_{j,b}\right\|_{L_1}\right)$$

which computes the $L1$ norm at corresponding rows over all matrices. Finally, the output

$$o(x_b^{(i)}) = \sum_{i=1}^{n} K_b(f(x^{(i)}))$$

is concatenated with $f(x)$ and provided as input for the next layer of the discriminator.

Next, we add two additional convolutional layers of kernel size 5×5 with batch normalization that take as input the concatenation of the four layer of convolution and the output of minibatch discrimination. The output of this final convolution is fed into a fully-connected linear layer that flattens output to dimension batch size (64). See Appendix for more information.

5.3 Metrics

In computational astrophysics, the way to evaluate generative models is to study their ability to reproduce the characteristic test statistics of the original dataset. We consider the following metrics.

- **Pixel Intensity:** A first-order statistic displayed as the average histogram of pixel values for a generated ensemble of images. We use the Kolmogorov-Smirnov test to compare similarity of cumulative distribution functions of pixel intensity against validation set.
- **Shear Power Spectrum:** A second-order statistic that is traditionally used to compare generated maps with real images. In general relativity, the matter density of the universe is defined by the power spectrum

$$\langle \widetilde{\delta}(l)\widetilde{\delta}^*(l')\rangle = (2\pi)^2\delta_D(l-l')P_\kappa(l)$$

  where ℓ is the Fourier mode. The generated graphs will plot the amplitudes of each Fourier component of the generated image. As in the CosmoGAN paper, we plot 248 Fourier modes from a batch and compare generated power spectrum to that of validation set.
- **Principal Components Analysis (PCA):** Novel metric highly unused in astrophysics literature. We implement PCA to approximate how well generated distribution resembled real distribution of images by plotting the first two principal eigenvectors for generated batch images at certain epochs. A converging distribution of points would give us a strong indication that the model is learning with stability.

6 Results and Discussion

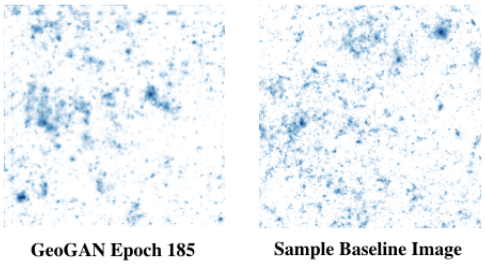


Figure 2: Sample generated image (left) vs. baseline image (right)

After training for 185 epochs on 16,000 images, we obtain a Kolmogorov-Smirnov test score $p = 0.989$, which is very close to the KS-score reported by Mustafa et. al. [2018] ($p = 0.999$). Validation and GeoGAN Pixel intensity histograms are almost identical, except for at tail behavior where GeoGAN shows slight oscillatory behavior. Observe that the point of discrepancy occurs when the pixel counts are very infrequent, so we would expect to not see discernible changes in the images by naked eye.

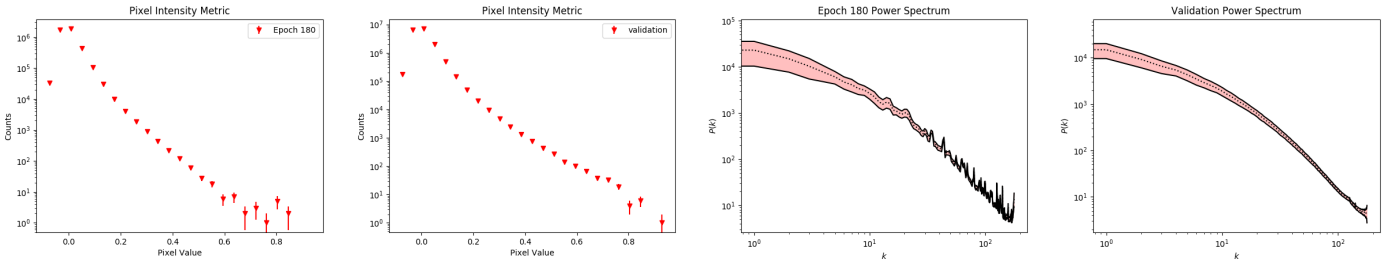
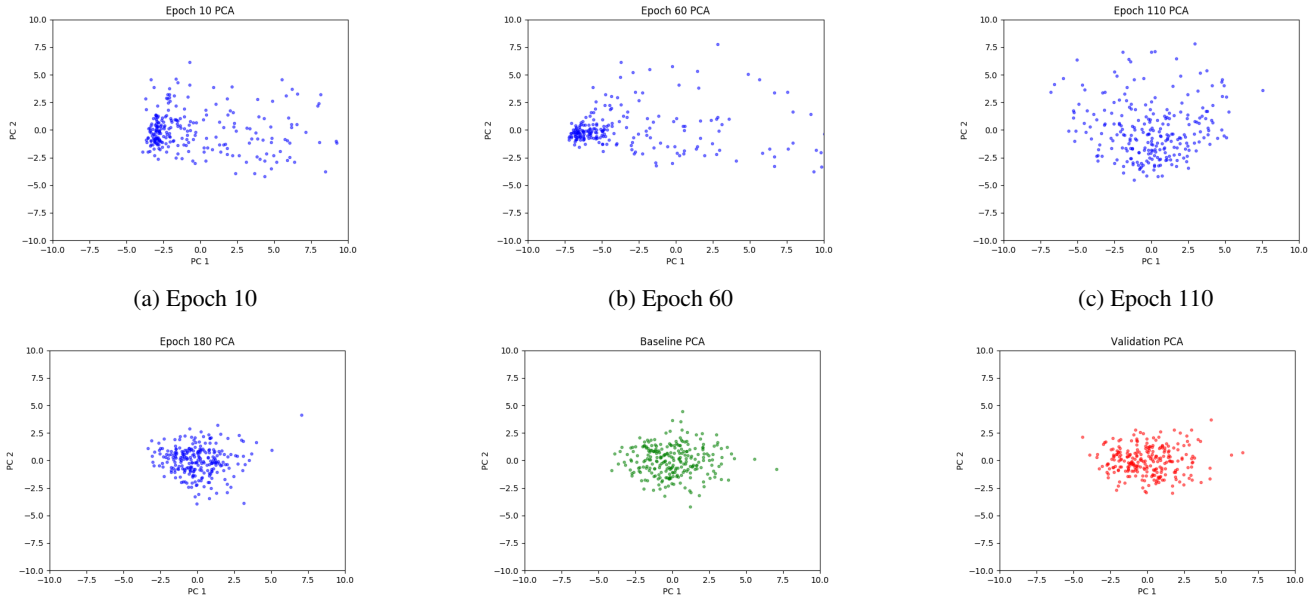


Figure 3: Results for Pixel Intensity and Power Spectrum

The GeoGAN loss curves begin to show converging behavior around epoch 100. At the beginning of training, the highly oscillatory behavior of the loss functions is characteristic of GANs, since when the generator is far from the real data distribution, the discriminator will penalize it, subsequently changing its loss. Apparent convergence of the GeoGAN is consistent with theoretical findings that the model will converge to Nash Equilibrium given large enough sample size. With more computational resources, training on more images can further pronounce this convergence.

Figure 4: Generator (Left) and Discriminator (Right) Losses

Principal Components Analysis gave us supplementary confidence that our model was indeed converging to the real probability distribution. Near the start of training, the model is very noisy, and near convergence it well-approximates Validation data.



(a) Epoch 10 (b) Epoch 60 (c) Epoch 110

(d) Epoch 180 (e) Baseline (f) Validation

Figure 5: Results for PCA

In the context of existing literature, our model can be viewed as a proof-of-concept of an alternative GAN architecture that can achieve strong results despite less access to information and compute. Although we trained on a smaller dataset compared to [7, 9], our $p = 0.989$ for the Kolmogorov-Smirnov test score differs by just 0.01 compared to models that were trained for four times longer and on a larger dataset[1] Additionally, whereas prior work outlined difficulties with increased instability on GANs when measuring performance with sensitive summary statistics, our model displayed a steadily decreasing loss that also did not run into common GAN issues such as mode collapse. This particularly highlights the stability that minibatch discrimination brought to our model, in having the discriminator learn about the similarities within the minibatch itself.

Additionally, whereas prior work employs common GAN performance metrics Frechet Inception Distance, we posit that our PCA-based metric provides a visual intuition for how well generated distributions resemble the real distribution of images. Compared to FID's reliance on calculating means and covariances of generated images, the simple action of plotting first two principal eigenvectors for generated batch images at certain epochs could continuously help computational cosmologists in the field detect anomalous model behavior earlier in training by inspection. Finally, our metric may aid in assessing when to stop training based on the shape of the PCA distribution, as prior work has noted that currently, the precision of the generator at the point we stop the training is completely arbitrary [7].

7 Conclusions and Future Work

In this paper, we submit two main contributions to the computational cosmology literature. First, we test a proof-of-concept for a geometric GAN trained with a hinge loss and demonstrate that it converges to similar test statistics as current state-of-the-art techniques in less time (1 day instead of 4 days). Our model also appears to have more stable behavior, achieving a steadily decreasing loss without mode collapse. Second, we submit a novel PCA-based metric, that has not been used in the astroimaging community, to provide visual intuition for future researchers looking to detect anomalous behavior in training.

There are many avenues of future work. With more time, we would have liked to explore more advanced metrics as test statistics such as Minkowski functionals, as well as generate correlation matrices to analyze statistical independence of generated samples. Finally, as prior work points out, the area where real progress will be made is assessing the ability of GANs to interpolate in the parameter space of physical models [7].

8 Contributions

Michel focused on algorithm optimization, by implementing GeoGAN loss, PCA, and minibatch discrmination in tensorflow. Amay focused primarily on implementation and training of the GAN, as well as integration with GCP and generation of plots. Andrew focused on implementing metrics to assess the performance of our models, including pixel intensity and power spectrum. Our code can be found here (https://github.com/amayagg/cosmogang).

[1] Due to computational resource limitations, training on the full dataset for four days was infeasible.

References

[1] Martin Arjovsky and Léon Bottou. *Towards Principled Methods for Training Generative Adversarial Networks*. 2017. eprint: arXiv:1701.04862.

[2] Carlo Giocoli, Marco Baldi, and Lauro Moscardini. "Weak lensing light-cones in modified gravity simulations with and without massive neutrinos". In: *Monthly Notices of the Royal Astronomical Society* 481.2 (Sept. 2018), pp. 2813–2828. ISSN: 1365-2966. DOI: 10.1093/mnras/sty2465. URL: http://dx.doi.org/10.1093/mnras/sty2465.

[3] Carlo Giocoli et al. "Fast weak-lensing simulations with halo model". In: *Monthly Notices of the Royal Astronomical Society* 470.3 (June 2017), pp. 3574–3590. ISSN: 1365-2966. DOI: 10.1093/mnras/stx1399. URL: http://dx.doi.org/10.1093/mnras/stx1399.

[4] Alexia Jolicoeur-Martineau. *The relativistic discriminator: a key element missing from standard GAN*. 2018. arXiv: 1807.00734 [cs.LG].

[5] Jae Hyun Lim and Jong Chul Ye. "Geometric GANs". In: (2017). eprint: arXiv:1705.02894.

[6] Seongkyu Mun et al. "GENERATIVE ADVERSARIAL NETWORK BASED ACOUSTIC SCENE TRAINING SET AUGMENTATION AND SELECTION USING SVM HYPER-PLANE". In: *Detection and Classification of Acoustic Scenes and Events* (2017). URL: https://pdfs.semanticscholar.org/cbe8/fa1b7d7d602049a186c9340fb46f8b791a23.pdf.

[7] Mustafa Mustafa et al. "CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks (code: https://github.com/MustafaMustafa/cosmoGAN)". In: *Computational Astrophysics and Cosmology* 6.1 (June 2019). DOI: 10.1186/s40668-019-0029-9.

[8] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2015. arXiv: 1511.06434 [cs.LG].

[9] Andres C. Rodríguez et al. "Fast cosmic web simulations with generative adversarial networks". In: *Computational Astrophysics and Cosmology* 5.1 (Nov. 2018). ISSN: 2197-7909. DOI: 10.1186/s40668-018-0026-4. URL: http://dx.doi.org/10.1186/s40668-018-0026-4.

9 Appendix

Following the instructor posts on Piazza (https://piazza.com/class/k0s9q710pjn29t?cid=1397) permitting Appendices past the 5-page limit, we are including topical algorithms.

Algorithm 1: Virtual Batch Normalization

Given reference batch R of size m **for** $i \leftarrow 0$ **to** *n* **do**
  given $B[i] = \{x^{(i)}\}_{i=1}^{i=m}$
  $\mu_B = \frac{1}{2m}\sum_{i=1}^{m} B[i] + R[i]$
  $\sigma_B^2 = \frac{1}{2m}\sum_{i=1}^{m}(B[i] - \mu_B)^2$
  $x_i \leftarrow \frac{x_i - \mu_B}{\sigma_B}$
end

Algorithm 2: Minibatch Discrimination

$f(x^{(i)}) \in R^n$ be intermediate layer output in D, $0 \leq i \leq n$
Initialize trainable $T \in R^{n \times m \times p} \sim N(0, 1)$
$M_i = T * f(x^{(i)})$, $0 \leq i \leq n$
for $b \leftarrow 0$ **to** *B* **do**
  for *all* $i, j \leftarrow 0$ **to** *n* **do**
    $K_b(f(x^{(i)}), f(x^{(j)})) = \exp\left(-\left\|M_{i,b} - M_{j,b}\right\|_{L_1}\right)$
  end
  $o(x_b^{(i)}) = \sum_{i=1}^{n} K_b(f(x^{(i)}))$
end
concatenate $o(\mathbf{x})$ to $f(\mathbf{x})$