GANDI: Generative Adversarial Networks for Detecting Irregularities

Ehud Karavani
Adviser: Dr. Matan Gavish

Abstract

We present a novel methodology to apply the generative adversarial networks (GANs) model to the problem of anomaly detection using the usually discarded discriminator as our detector. While training, the discriminator learns some meaningful representation of the true distribution, otherwise it's generator partner would not improve it's own generating abilities. We try to identify the point where the discriminator holds the best possible representation of the true distribution and harness it for the task of detecting irregularities.

Introduction

Generative Adversarial Networks

Generative models are models that can learn to create synthetic data that is similar to data that is given to them. Over the years, many different models tackled this problem, but recently, one promising approach devised is Generative Adversarial Networks (which we abbreviate as GANs) [6] that has dramatically sharpened the possibility of AI-generated content. GANs manage to skip over the difficulty of approximating many intractable probabilistic computations that arise in maximum likelihood related strategies by using an adversarial scheme and harnessing the expressive power of neural networks [5].

We do not attempt to do a complete survey of GANs, but only to simply describe it in a non-rigorous, but hopefully intuitive, way. A GAN model is composed of two computational components (mainly, neural nets) - a generator (denoted as G) and a discriminator (denoted as D). Given a distribution P (e.g. a data set) that we'd like to learn (to be able to generate more data similar to it), the two nets play a game. The generator, G, inputs a random noise seed $z \sim \eta$ and outputs a synthetic generated sample $G(z)$. The discriminator, D, then receives two inputs - the generated sample $G(z)$ and a true sample x drawn from the true distribution P. It is then the job of the discriminator to be able to tell which input is synthetic and which is real. An intuitive explanation for the process of learning is that the better the discriminator discriminates - the better G has to become generating samples that resemble true data, and the better G becomes (outputting real-like data) - the better D has to be in discriminating between true and fake.

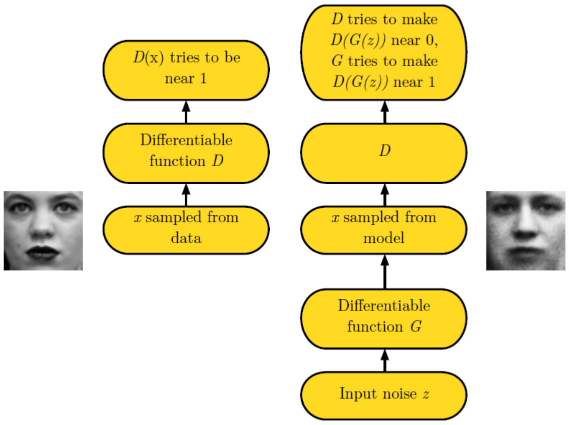
If we'll define the discriminator's loss function as standard cross-entropy:

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{x\sim P}\left[\log\left(D\left(x\right)\right)\right] - \frac{1}{2}\mathbb{E}_{z\sim\eta}\left[\log\left(1 - D\left(G\left(z\right)\right)\right)\right] \tag{1}$$

And Then define the generator's loss to be:

$$J^{(G)} = -J^{(D)} \tag{2}$$

The competition above can be formally described as a zero-sum minimax game. Hence, the desired pa-



D(x) tries to be near 1
Differentiable function D
x sampled from data
D tries to make $D(G(z))$ near 0, G tries to make $D(G(z))$ near 1
D
x sampled from model
Differentiable function G
Input noise z

Figure 1: The GAN framework. Taken from [5].

rameterization for G can be achieved by solving for:

$$\theta^{(G)*} = \arg\min_{\theta^{(G)}} \max_{\theta^{(D)}} V\left(\theta^{(D)}, \theta^{(G)}\right) \tag{3}$$

Where V is the value function and $\theta^{(D)}, \theta^{(G)}$ are the models parameterization (i.e. the net's weights).

Under that formalization, *Nash equilibrium* is achieved (assuming infinite computational power) when the generator produces samples identical to the samples from the true distribution and the discriminator is left confused, outputting answers at random (since it sees two identical inputs that one is labeled with 0 and the other labeled 1).

The framework can be visually described in 1.

Anomaly Detection

Anomaly detection refers to the problem of finding patterns in data that do not conform to a well defined notion of normal or expected behavior. These non-conforming patterns go by many names like anomalies, outliers, exceptions, to name a few, and vary dependent on the application domains. The goal of anomaly detection is to identify these irregular data points in a given sample and report them. Flagging out outliers is a challenging task for several reasons [4]

1. Lack of labeled data for training and validation.
2. Noisy data can be easily regarded as anomalies and it is often difficult to distinguish and adjust.
3. defining a normal region which encompasses every possible normal behavior is very difficult.
4. The exact notion of anomaly is different in different domains.
5. Normal behavior can be an evolving thing, and whatever learned now might not be sufficient for future behavior.

Points (1) and (2) are due to data gathering and measurement and are not in the scope of current work. We argue our method might solve for points (3) and (4), and future work involving online-learning in a similar framework might solve for (5). Where point (4) can be adjusted by using different NN architectures borrowed from the relevant domain, and point (3) is the basic premise of this work.

The Basic Premise

As stated, the number of possible anomalies can't be accounted for. Thus, it is impossible to train a discriminative model to tell if a given datum is normally behaved or is it discordant. Hence, a generative model needs to come to the rescue. A model that knows the distribution P can easily output a detectable different output when given $\alpha \notin P$ that will alert us that α is not an expected datum. This is exactly where we can harness our discriminator D.

The hypothesis goes as such - at some point in our training, D outputs values near 1 when encountered with true data and near 0 for data impersonated to be true data. If we could pause the training at that exact point, we would have a machinery that outputs ~ 1 for normal behavior and outputs some other value (that we need to know to identify and differentiate) for data that is not normally behaved. That is, we solve for challenge (3) in creating a very generalized anomaly detector.

Moreover, given the vast computational power that current state-of-the-art neural networks posses, we can solve challenge (4) by designing different NN for different applications, borrowing the results of an already prolific research harnessing NN for these applications. While more classical methods will probably need knowledge based transformation to better represent the data in order to perform, NN are (almost) agnostic about data preprocessing and can use their power to learn some internal representation while at it. A similar example for this trend is the shift that occurred in the field of computer vision where hand-crafted preprocessed representations (such as histogram of gradients or frequency filtering) and classical machine learning algorithms gave away to raw inputs and expressive NN. Thanks to NN, this "out-of-the-box" behavior, makes GANDI both powerful and easy to use, and can make the process of detecting of anomalies much more accessible in many applications.

Theoretical Challenges and Practical Overcomes

As stated above, all the applications of GANs known to the writer deal with data generation. That is, at the end of the training process they discard the discriminator and use the generator to produce more data for their need. This method of operation is supported by the theory too, as the Nash equilibrium sentences our discriminator to life of absolute confusion at which its discarding may seem more like euthanasia.

Since I lack the skills to twist the theory behind the model so that our discriminator will win the game (i.e. discriminate immaculately) and the generator to lose (i.e. generate noise), I turn to the empirical process of finding that sweet-spot during training where the discriminator acts it's best. I can then pause training and export (or rather metamorphose) our discriminator as an anomaly detector. This is based on the notion that the discriminator fitness is parabolic - it begins knowing nothing (randomly initialized) and ends up confused (being the generator winning the competition and thus feeding the discriminator with two identical inputs which are differently labeled). However, in between these two ends, we know it learn some meaningful representation of the problem of identifying true data from false one; otherwise it wouldn't be able to contribute to the performance improvement of its generator foe - which we know to improve for sure.

Methods

First step is to try the above hypothesis on a simple, easy to validate, easy to characterize problem. The toy problem chosen for the mission is a simple 1-dimensional Gaussian distribution. Gaussian distribution can be easily identifiable when learned (i.e. it has low entropy structure). The low dimension makes it easy to check, both visually and analytically, using vastly available statistical approaches, how good the generator is learning the true distribution. The choice of numerical distribution, instead of a data set, allows us to draw as many samples as needed while characterizing the learning process. Moreover, it allows us to easily define anomalies by specifying another (Gaussian) distribution with different parameters. The performance of the anomaly detectors can be compared with the effect size between the two distributions:

$$\frac{\mu_{true} - \mu_{anomaly}}{\sigma_{true} \cdot \sigma_{anomaly}} \tag{4}$$

As where in our case, we fix $\sigma_{true} = \sigma_{anomaly} = 1$ and thus the effect size is only the shift in means. A good sanity check is that the bigger the effect - the better the detection performance.

Under these specifications, the hypothesis can be rephrased: Can we use various goodness-of-fit measurements applied to G during training that will reveal when is it best to stop the process of training and say the current discriminator is the best anomaly detector can be achieved in the process?

To test that hypothesis, a Python framework was constructed using TensorFlow. The experiment initializes a GAN model of two neural network, different architectures and hyperparametrization (in both nets) were tried. The training process is then paused every k steps (or otherwise specified) and it takes several measurements of the system. Mainly it tests the performance of the discriminator as an anomaly detector and, in addition, it performs several goodness-of-fit tests between the generator and the true distribution. The tests done were Kullback-Leibler divergence [8], Kolmogorov-Smirnov statistic and Kolmogorov-Smirnov p-value [7, 11], Anderson-Darling [1, 2], total-variation distance (ℓ_1 difference between the two cumulative density functions (CDFs)) and visual inspections of the CDFs, PDFs and QQ-plot (quantile-quantile plot [12]) between the generated and true distribution over time.

We tried to characterize the behavior of how these measurements correspond to the measurements of detection accuracy as a function of training iteration and the training loss progression.

We test the performance of the anomaly detector as if it was a binary classification problem, supplying it with samples from the true distribution labeled as 1 (corresponding the value D tried to achieve on real samples while training) and samples from the anomaly distribution labeled with 0. We can then apply several metrics on the resulting contingency matrix and we mostly use the area under the curve (AUC).

To avoid generator mode collapse we equipped our discriminator with a deterministic *minibatch discrimination* layer that "punishes" the generator for lack of diversity. To assist convergence we used cross-entropy loss (rather than the min-max game) since gradient descent is not designed to find Nash equilibrium, but some low value cost function instead [10].

Results

Training converges and it seems the generator does perform better the longer the train goes #FIG-GIF_of_(cdf, pdf, qq)#. However, there is no apparent relation between the loss, the number of iterations and the goodness of fit statistics. These do improve as training progresses, but also have an oscillating behavior after they seem to converge Fig 2.

This lack of monotonicity in the behavior of the GAN model is, unfortunately, also present in D's behavior as an anomaly detector. When observing the AUC performance of the anomaly detector through time Fig 3 and when observing it as a function of the various goodness-of-fit measurements Fig 4.

This non-monotonicity is shown between different effect-sized of anomalies as well, i.e. small effect-sizes having better AUC than larger effect-sizes, but these were rare and were solved when introducing more powerful generator net and providing reasonable training time.

In addition, the single-value AUC, does not capture the shape of the ROC curve which was sometime peculiarly shaped more as a sigmoid, rather then the classic over-the-diagonal Fig 5. We could not determine if this behavior is due to some computational artifact or due to inherent behavior of the model caused by its design (e.g. architecture or hyperparameters).

Discussion

We presented a novel method for doing anomaly detection using the usually discarded discriminator of the GAN model. At its basis stands an idea for general and versatile anomaly detector. However, the process of achieving it relays on good intuition and somewhat poor theoretical assumptions and thus we turned to empirical experiments. We could not characterize the behavior of this model as an anomaly detector and could not correlate its performance to any measurable estimands. Specifically, we did not find it neatly correlated with the improvement of the generator or the loss function.

The oscillated behavior might be due process of peak performance - the generator being good confuses the discriminator which, in turn, falls behind. This low performance of the discriminator causes the generator to atrophy. The declined performance of the generator make it easier for the discriminator to discriminate and, thus, to improve again. The better the discriminator - the better the generator becomes. And we're back to the point where we started. We remind that gradient descent based optimization will probably not achieve equilibrium due to the non-convexity of the problem, but rather some other steady-state.

One possible way to combat this oscillating behavior of the model (appearing in both goodness-of-fit statistics and loss values) might be through the the use of a smoother loss function like Earth-Mover Distance (Wasserstein distance) as suggested in [3]. Let-

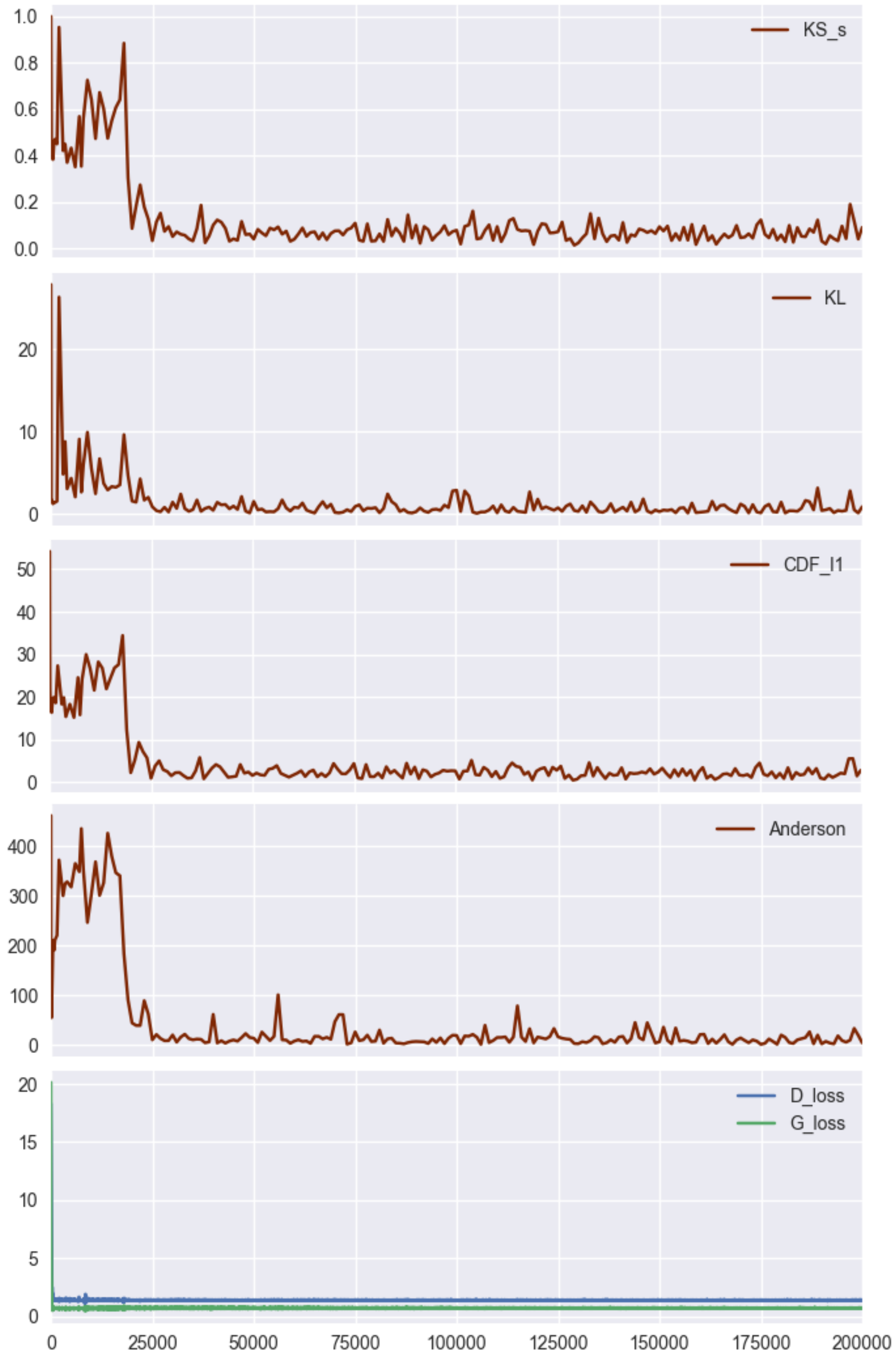


Figure 2: The behavior of the loss and goodness-of-fit tests (top to bottom: Kolmogorov-Smirvov statistic, KL divergence, total variation, Anderson-Darling) during training.

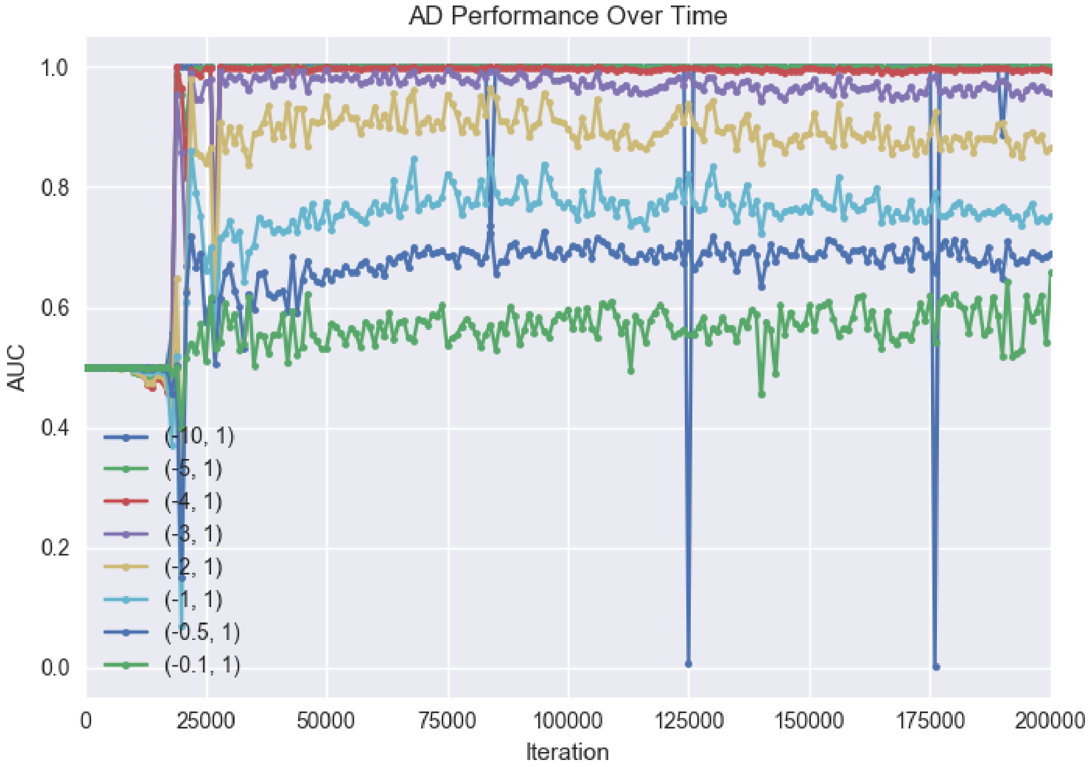


Figure 3: Anomaly detector shows oscillating and non-monotonic behavior as training progress.

ting go of the cross-entropy might produce more stable results.

The presented experiment cannot be naturally scaled as is, because it is not trivial to test for goodness-of-fit in high-dimensional data. This is because distance between two high-dimensional CDFs is not well defined unless considering all possible combinations (of dimensions) which makes it exponentially hard. One resort is to apply *classifier two-sample test* [9] to the generated and the true data; but, like high-dimensional approximations for CDF-based statistics, there is no one gold-standard way to do so.

Finally, We must remember that GAN models were derived for the sake of their generator. No fundamental work was done on the model's discriminator since it is always discarded after training. We hope this work can convince that there might be a good use for the discriminator as well; as the say goes: one's trash is another's treasure.

Acknowledgment

The author would thanks Dr. Matan Gavish for his bright ideas and helpful insights through the process of this research.

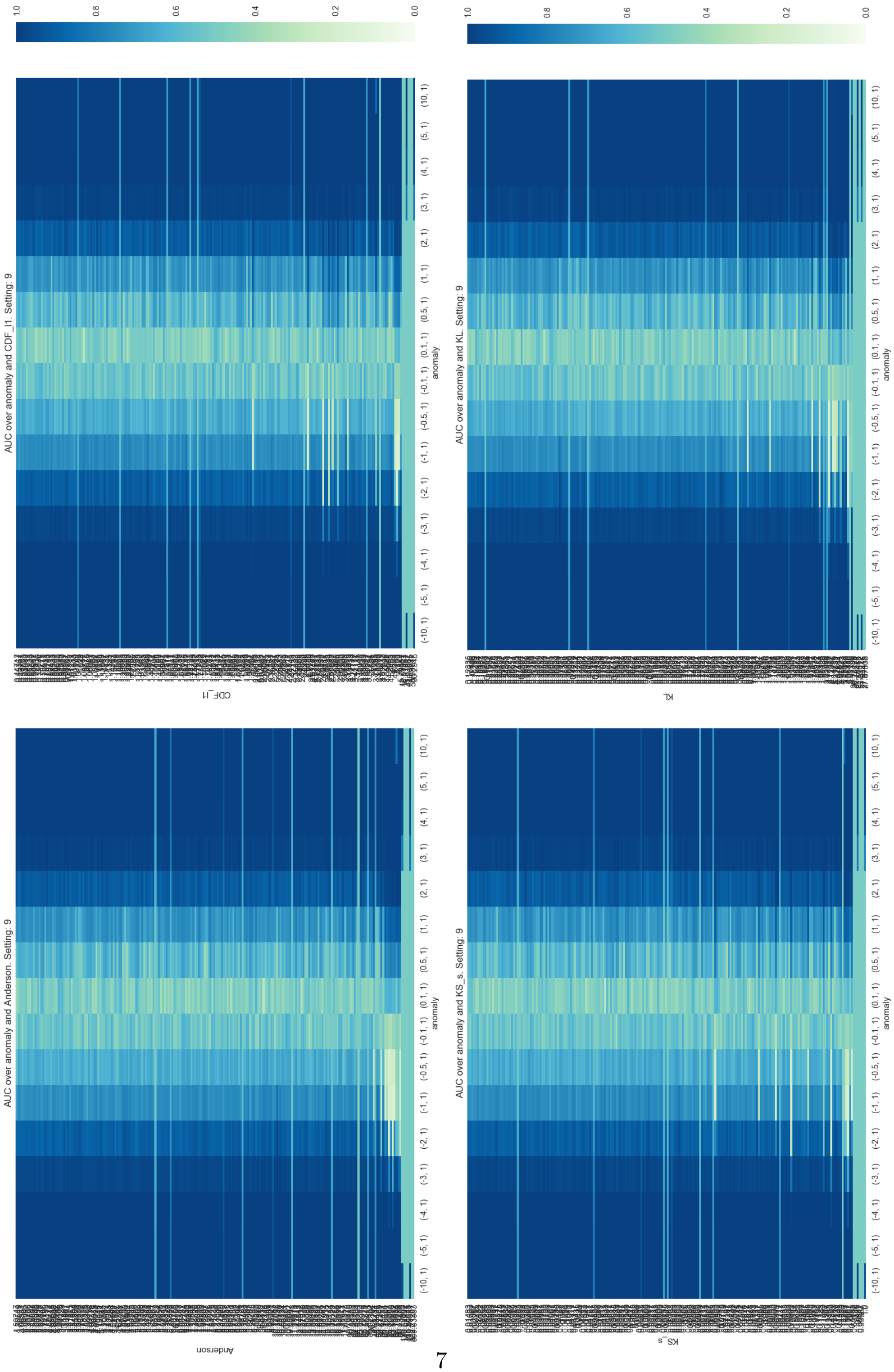


Figure 4: Heatmaps presenting the performance of the anomaly detector as a function of the goodness-of-fit statistic measured at the same training iteration (y axis) and over different anomalies (x axis). It can be observed that there is no monotonic relation (the bright horizontal lines) between the anomaly detector performance and the performance of the generator. Goodness of fit statistics (clockwise): total variation, KL divergence, KS, Anderson-Darling.

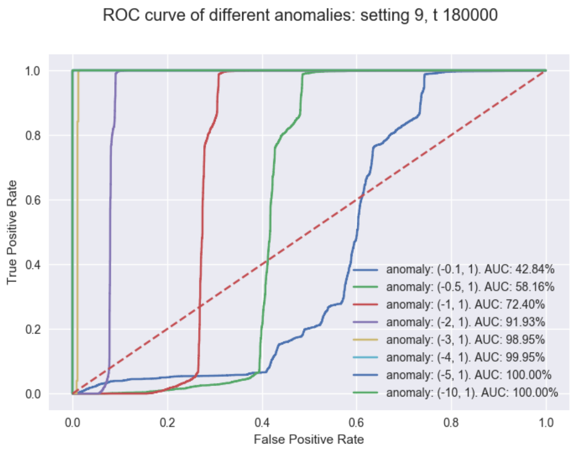


Figure 5: An example of sigmoidal ROC curve at iteration 180k.

References

[1] T. W. Anderson and D. A. Darling, *Asymptotic theory of certain 'goodness of fit' criteria based on stochastic processes*, Ann. Math. Statist. **23** (1952), no. 2, 193–212.

[2] T. W. Anderson and D. A. Darling, *A test of goodness of fit*, Journal of the American Statistical Association **49** (1954), no. 268, 765–769.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou, *Wasserstein gan*, arXiv preprint arXiv:1701.07875 (2017).

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar, *Anomaly detection: A survey*, ACM computing surveys (CSUR) **41** (2009), no. 3, 15.

[5] Ian Goodfellow, *Nips 2016 tutorial: Generative adversarial networks*, arXiv preprint arXiv:1701.00160 (2016).

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, *Generative adversarial nets*, Advances in neural information processing systems, 2014, pp. 2672–2680.

[7] Andrey Kolmogorov, *Sulla determinazione empirica di una lgge di distribuzione*, Inst. Ital. Attuari, Giorn. **4** (1933), 83–91.

[8] Solomon Kullback and Richard A Leibler, *On information and sufficiency*, The annals of mathematical statistics **22** (1951), no. 1, 79–86.

[9] D. Lopez-Paz and M. Oquab, *Revisiting Classifier Two-Sample Tests*, ArXiv e-prints (2016).

[10] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, *Improved techniques for training gans*, Advances in Neural Information Processing Systems, 2016, pp. 2234–2242.

[11] N. Smirnov, *Table for estimating the goodness of fit of empirical distributions*, Ann. Math. Statist. **19** (1948), no. 2, 279–281.

[12] M. B. Wilk and R. Gnanadesikan, *Probability plotting methods for the analysis for the analysis of data*, Biometrika **55** (1968), no. 1, 1–17.