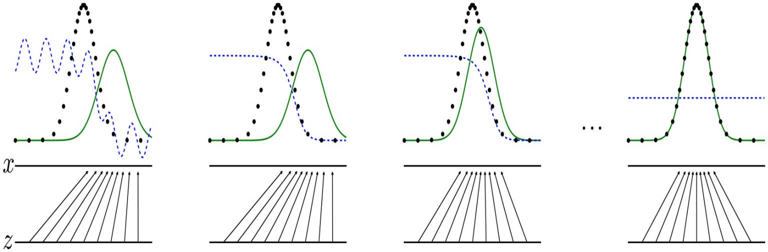
Karlsruhe Institute of Technology

Generative Adversarial Networks for Outlier Detection

Applied and Algorithmic Views on Machine Learning, Figure below from [Goo+14]

Leander Kurscheidt | August 1, 2017

INSTITUTE FOR PROGRAM STRUCTURES AND DATA ORGANIZATION CHAIR FOR SYSTEMS OF INFORMATIONMANAGEMENT



Introduction to generative models

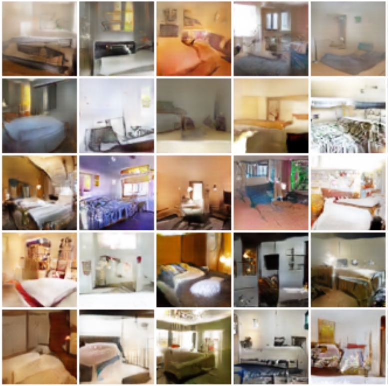


Figure: Samples from a Wasserstein-GAN, Selection from a figure from [ACB17]

Introduction to generative models

In our context an (optimal) generative model is a function

$$G : P_z \to P_{data}$$

where P_z is an arbitrary distribution, often called noise-distribution. P_{data} is the distribution we want to sample from.

Generative Adversarial Networks

as formulated by [Goo+14]:

discriminative model $D(x; \theta_d)$
tries to:

- assign 1 to elements of the original data
- assign 0 to elements produced by the generator
- *"tries distinguish real and generated samples"*

generative model $G(z; \theta_g)$
tries to:

- maximize $D(G(z))$
- *"tries to generate samples that fool the discriminator"*

GANs converge to a Nash-Equilibrium, as shown by [Heu+17].

Generative Adversarial Networks

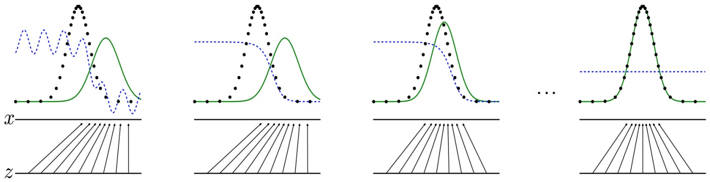


Figure: Generative Adversarial Networks over time, Figure from [Goo+14]

Generative Adversarial Networks

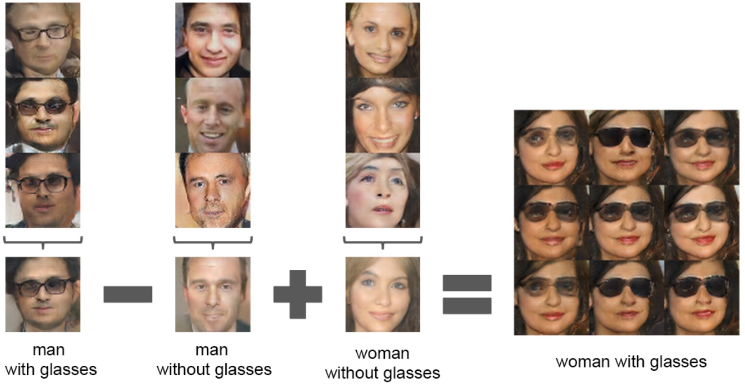


Figure: vector arithmetic on the generators input, Figure from [RMC15]

Generative Adversarial Networks

Figure: Learning cross domain relations between shoes and handbags, Figure from [Kim+17]

Comparison to usual neural networks

the traditional approach:
While other approaches exist (for example variational autoencoders [KW13]), most of the applications of neural networks are of an discriminative nature, for example classification. Generative models open up exciting new possibilites.

Laying the groundwork

- Deep Convolutional GANs [RMC15]
- Improved Techniques for Training GANs [Sal+16]

Two fundamental papers that contain a lot of advice on which architecture and parameters to choose.

Utilizing the noise vector

supervised: Conditional GANs [MO14]

- via explicit vector y, that both generator and discriminator recieve.
- y can later be used to control the properties of the generated samples.

unsupervised: InfoGAN [Che+16]

- splits the generators input vector z into n (the noise) and c (encoded features).
- c gets ignored when maximizing $D(G(z))$
- additional penalty enforces information-theoretic relationship between c and $G(n, c)$.
- properties of c have to be experimentally discovered.
- hard to get right

Preventing mode collapse

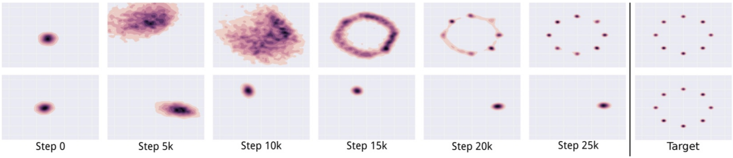


Figure: Figure from [Met+16]

Unrolled GANs [Met+16] introduced some divergence-independent methods to deal with mode-collapse at a performance cost.

A problematic divergence

(here GAN is used to refer to GANs as formulated by [Goo+14])

- training GANs is hard because the resulting quality varies (unable to just train until convergence)
- training GANs is hard because avoiding mode-collapse, stationary orbit etc. requires delicate balancing between the generator and the discriminator
- there are some theoretical reasons why this is happening (and why minimizing *JSD* is flawed)
- *JSD* is a divergence based on mutual information, and [AB17] showed that for arbitrary small perbutations on two manifolds a perfect discriminator is always existing (and *JSD* is maxed out) leading to vanishing gradients.

Other divergences

Recent research focused on selecting more stable divergence/metrics that don’t rely on relative probability (that produce sensible gradients for distributions that are close, but not overlapping) and translating them into algorithms. Examples are the Wasserstein (Or Earth-Mover’s) distance [ACB17], improved by [Gul+17], or the Cramer-distance [Bel+17].

Using GANs for outlier-detection

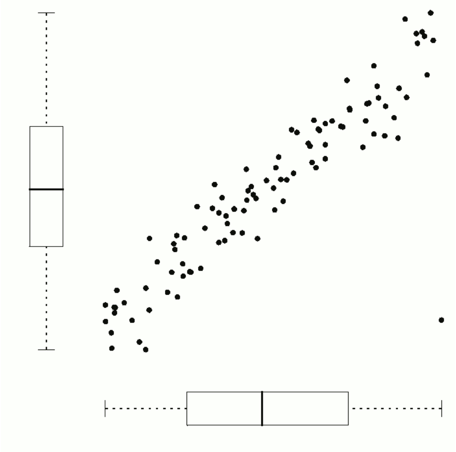


Figure: By Sigbert - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=8271928

Roadblocks

Issues when using GANs, as formulated by [Goo+14], for outlier-detection

Discriminator Convergence

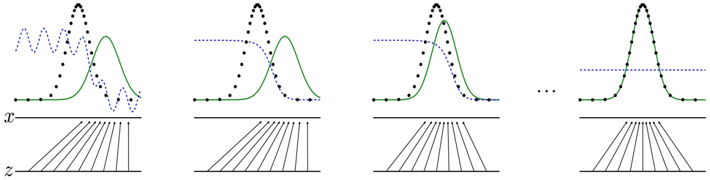


Figure: Generative Adversarial Networks over time, Figure from [Goo+14]

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)} \wedge P_g = P_{data} \implies D_G^*(x) = \frac{1}{2}$$

A similar problem exists for Wasserstein- & Cramer-GANs.

Overfitting

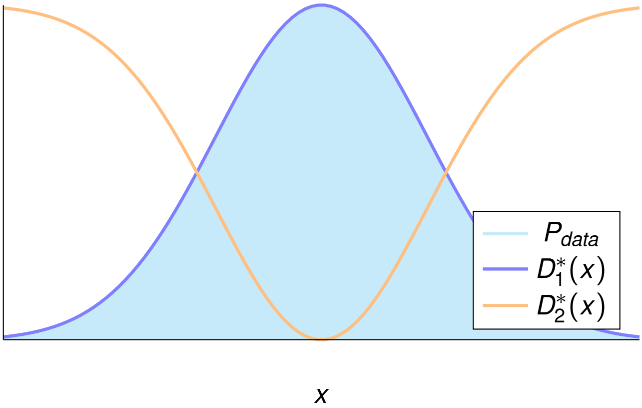
The theoretical analysis in [AB17] showed that GANs, as formulated by [Goo+14], are massivly overfitting on P_{data}.

Objective

An architecture that converges towards $D^*(x) = 1$ for $x \in P_{data}$ and $D^*(y) = 0$ for $y \notin P_{data}$

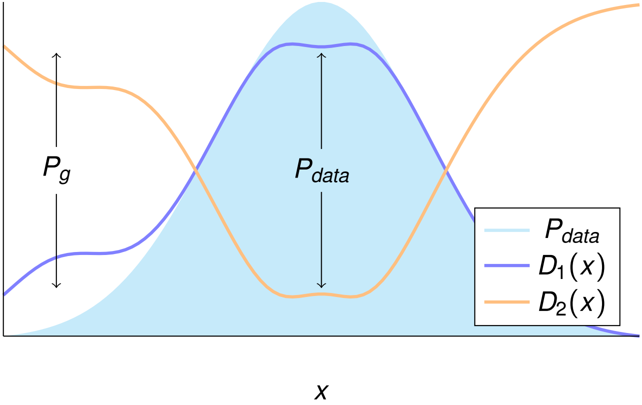
Outlier-GAN Idea

What we want to converge to:



Outlier-GAN Idea

How we converge:



Outlier-Gan Details

Karlsruhe Institute of Technology

$$F_G(z, \theta_G) = \mathbb{E}_{z \sim p_z}[D_1(G(z))D_2(G(z)) + D_2(G(z))]$$
$$= \mathbb{E}_{x \sim p_g}[D_2(x)(1 + D_1(x))]$$

- high reward for D_1 and D_2 being both high
- converges towards P_g having a disjunct support from P_{data} (though support of P_g might be small, as observed by [AZ17] for JSD-Divergences)
- straightforeward extensions to address certain flaws
- only a idea sketch

Conclusion

- GANs add a new tool to the ML-toolbox
- recent theoretic breakthroughs make GANs more practical
- While Outlier-Gan is a nice idea, there are theoretical flaws (similiar ot the ones for the GANs as formulated by [Goo+14]). But I am optimistic that the solution can be adapted, so that this idea might provide a gradient towards a more scalable solution. It also might just work on simpler data.

Theoretical Results

GANs, as formulated by [Goo+14], minimize:

$$C(G) = -log(4) + 2 \times JSD(P_{data}||P_g)$$
$$= -log(4) + 2 \times (\frac{1}{2}KL(P_{data}||B) + \frac{1}{2}KL(B||P_{data}))$$

$$P_g = G(P_z)$$
$$B = \frac{1}{2}(P_{data} + P_g)$$

JSD is the Jensen-Shannon divergence

KL is the Kullback-Leibler divergence

References I

Martín Arjovsky and Léon Bottou. “Towards Principled Methods for Training Generative Adversarial Networks”. In: *CoRR* abs/1701.04862 (2017). URL: http://arxiv.org/abs/1701.04862.

Martín Arjovsky, Soumith Chintala, and Léon Bottou. “Wasserstein GAN”. In: *CoRR* abs/1701.07875 (2017). URL: http://arxiv.org/abs/1701.07875.

Sanjeev Arora and Yi Zhang. “Do GANs actually learn the distribution? An empirical study”. In: *CoRR* abs/1706.08224 (2017). URL: http://arxiv.org/abs/1706.08224.

Marc G. Bellemare et al. “The Cramer Distance as a Solution to Biased Wasserstein Gradients”. In: *CoRR* abs/1705.10743 (2017). URL: http://arxiv.org/abs/1705.10743.

References II

Xi Chen et al. “InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets”. In: *CoRR* abs/1606.03657 (2016). URL: http://arxiv.org/abs/1606.03657.

Ian J. Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. Ed. by Zoubin Ghahramani et al. 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.

Ishaan Gulrajani et al. “Improved Training of Wasserstein GANs”. In: *CoRR* abs/1704.00028 (2017). URL: http://arxiv.org/abs/1704.00028.

References III

Martin Heusel et al. “GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium”. In: *CoRR* abs/1706.08500 (2017). URL: http://arxiv.org/abs/1706.08500.

Taeksoo Kim et al. “Learning to Discover Cross-Domain Relations with Generative Adversarial Networks”. In: *CoRR* abs/1703.05192 (2017). URL: http://arxiv.org/abs/1703.05192.

Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *CoRR* abs/1312.6114 (2013). URL: http://arxiv.org/abs/1312.6114.

Luke Metz et al. “Unrolled Generative Adversarial Networks”. In: *CoRR* abs/1611.02163 (2016). URL: http://arxiv.org/abs/1611.02163.

References IV

Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: *CoRR* abs/1411.1784 (2014). URL: `http://arxiv.org/abs/1411.1784`.

Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *CoRR* abs/1511.06434 (2015). URL: `http://arxiv.org/abs/1511.06434`.

Tim Salimans et al. "Improved Techniques for Training GANs". In: *CoRR* abs/1606.03498 (2016). URL: `http://arxiv.org/abs/1606.03498`.