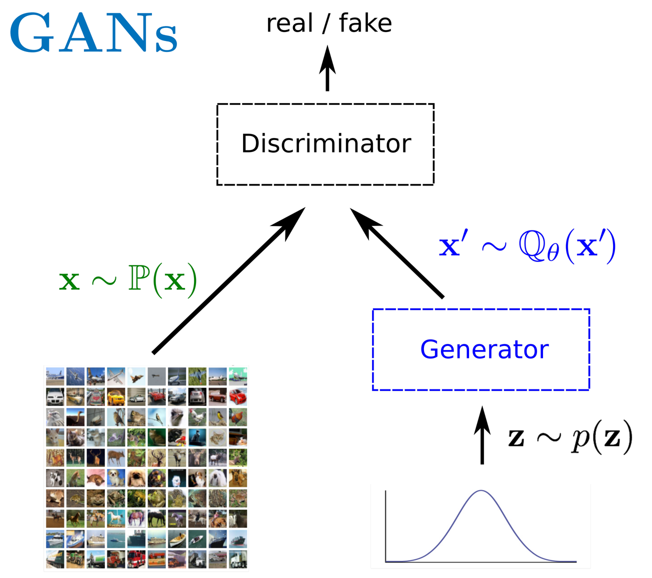
Stabilizing Training of Generative Adversarial Networks through Regularization

Kevin Roth   Aurélien Lucchi   Sebastian Nowozin   Thomas Hofmann

Institute of Machine Learning, ETH Zürich, Switzerland

Microsoft Research, Cambridge, UK

GANs



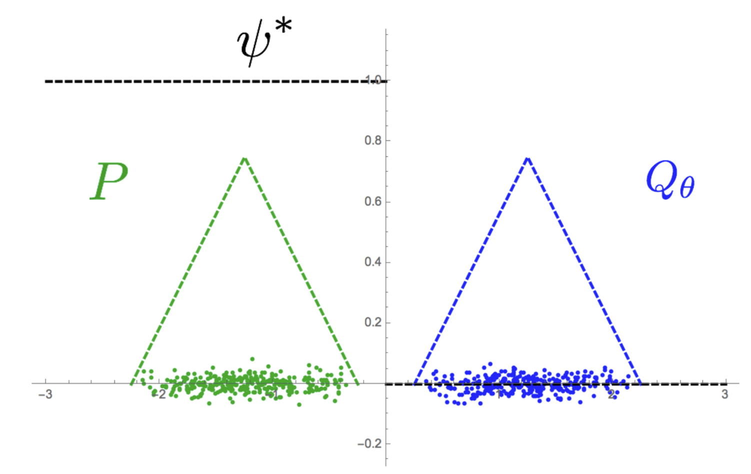
GANs are exciting, but ...
they're also notoriously hard to train!

GAN objective [2]

$$\min_G \max_\varphi \mathbf{E}_{\mathbb{P}}[\ln \varphi(x)] + \mathbf{E}_{p(z)}[\ln(1-\varphi(G(z)))]$$

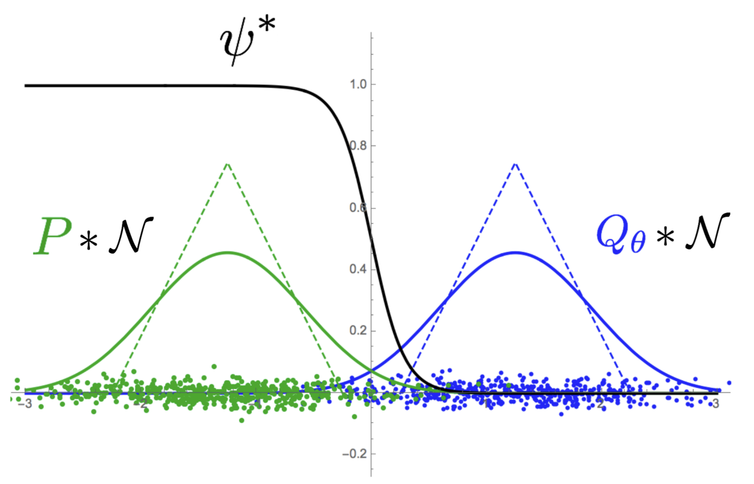
$$= \min_G D_{\mathrm{JS}}(\mathbb{P}\|\mathbb{Q}_\theta) \quad \text{for Bayes-optimal } \varphi^*(x)$$

Problem: Dim. Mismatch or $\mathrm{supp}(\mathbb{P}) \cap \mathrm{supp}(\mathbb{Q}_\theta) = \varnothing$



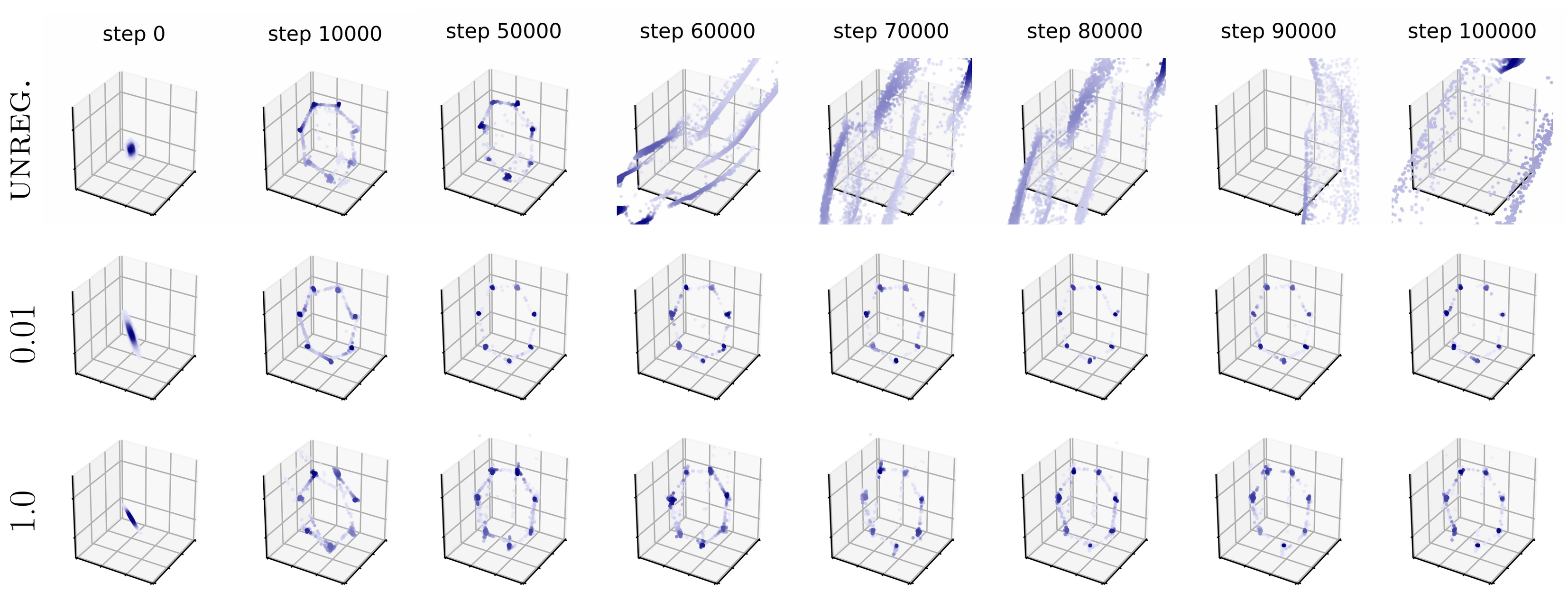
=> Undef. f-div. $D_f(\mathbb{P}\|\mathbb{Q}_\theta)$ =???

Solution: Adding Noise (Convolving Densities)



=> Regularizing Discriminator

Dimensionally Misspecified Submanifold Mixture



Unstable unregularized GAN vs. stable regularized GANs for different levels of γ.
The regularized GAN can essentially be trained indefinitely without collapse.

References

[1] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*, volume 2016, 2017.

[2] C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *NIPS*, 2014.

[4] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[5] S. Nowozin, B. Cseke, and R. Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pages 271–279, 2016.

[6] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

Training with Noise

From f-divergences to f-GAN objectives [4]

$$D_f(\mathbb{P}\|\mathbb{Q}) \geq \sup_{\psi \in \Psi} \left[F(\mathbb{P},\mathbb{Q};\psi) := \mathbf{E}_{\mathbb{P}}[\psi] - \mathbf{E}_{\mathbb{Q}}[f^c \circ \psi] \right]$$

- Practitioner: **Explicitly adding noise** $\boldsymbol{\xi} \sim \Lambda_\gamma \equiv \mathcal{N}(0, \gamma \mathbb{I})$ to $\mathbf{x} \sim \mathbb{P}, \mathbb{Q}$
- Theory: **Convolving Distributions**

$$\mathbf{E}_{\mathbb{P}}\mathbf{E}_{\Lambda}[h(\psi(\mathbf{x}+\boldsymbol{\xi}))] = \int h(\psi(\mathbf{x})) \int p(\mathbf{x}-\boldsymbol{\xi})\lambda(\boldsymbol{\xi}) d\boldsymbol{\xi}\, d\mathbf{x} = \mathbf{E}_{\mathbb{P}*\Lambda}[h \circ \psi]$$

=> **Convolved f-GAN Objective**

$$F(\mathbb{P} * \Lambda_\gamma, \mathbb{Q} * \Lambda_\gamma; \psi) = \mathbf{E}_{\mathbb{P}*\Lambda_\gamma}[\psi] - \mathbf{E}_{\mathbb{Q}*\Lambda_\gamma}[f^c \circ \psi]$$

Analytic Approximation

- For small noise variance γ we can Taylor expand ψ around $\boldsymbol{\xi} = 0$ [1]:

$$\psi(\mathbf{x}+\boldsymbol{\xi}) = \psi(\mathbf{x}) + [\nabla \psi(\mathbf{x})]^T \boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\xi}^T [\nabla^2 \psi(\mathbf{x})]\, \boldsymbol{\xi} + \mathcal{O}(\boldsymbol{\xi}^3)$$

- Third-order approximation (in $\boldsymbol{\xi}$) of F_γ via $F = F_0$ plus a correction, i.e.

$$F_\gamma(\mathbb{P},\mathbb{Q};\psi) = F(\mathbb{P},\mathbb{Q};\psi) + \frac{\gamma}{2}\left\{ \mathbf{E}_{\mathbb{P}}[\triangle \psi] - \mathbf{E}_{\mathbb{Q}}[\triangle(f^c \circ \psi)] \right\} + \mathcal{O}(\gamma^2)$$

- Interpretation: Laplace $\triangle = \mathrm{Tr}(\nabla^2)$ measures how much ψ and $f^c \circ \psi$ differ from their local average

Efficient Gradient-Based Regularization

- Chain-rule: $\quad \triangle(f^c \circ \psi) = \left(f^{c\prime\prime} \circ \psi\right) \cdot \|\nabla \psi\|^2 + \left(f^{c\prime} \circ \psi\right) \triangle \psi$
- Property of optimal discriminant ψ^* [3]: $\quad (f^{c\prime} \circ \psi^*)\, d\mathbb{Q} = d\mathbb{P}$

=> **Convenient cancellation** at $\psi = \psi^* + \mathcal{O}(\gamma)$ [1]:

$$\mathbf{E}_{\mathbb{P}}[\triangle \psi^*] - \mathbf{E}_{\mathbb{Q}}[\triangle(f^c \circ \psi^*)] = -\mathbf{E}_{\mathbb{Q}}[(f^{c\prime\prime} \circ \psi^*) \cdot \|\nabla \psi^*\|^2]$$

=> Tractable regularization which avoids (i) detrimental sampling variance, (ii) explicitly convolving the distributions, and (iii) the computation of Laplacians

Regularized f-GAN

$$F_\gamma(\mathbb{P},\mathbb{Q};\psi) = F(\mathbb{P},\mathbb{Q};\psi) - \frac{\gamma}{2}\Omega_f(\mathbb{Q};\psi), \quad \Omega_f(\mathbb{Q};\psi) := \mathbf{E}_{\mathbb{Q}}\left[(f^{c\prime\prime} \circ \psi) \cdot \|\nabla \psi\|^2 \right]$$

$$F_{\mathrm{JS}_\gamma}(\mathbb{P},\mathbb{Q};\varphi) = \mathbf{E}_{\mathbb{P}}[\ln(\varphi)] + \mathbf{E}_{\mathbb{Q}}[\ln(1-\varphi)] - \frac{\gamma}{2}\Omega_{\mathrm{JS}}(\mathbb{P},\mathbb{Q};\varphi)$$

$$\Omega_{\mathrm{JS}}(\mathbb{P},\mathbb{Q};\varphi) := \mathbf{E}_{\mathbb{P}}\left[(1-\varphi)^2 \|\nabla \phi\|^2\right] + \mathbf{E}_{\mathbb{Q}}\left[\varphi^2 \|\nabla \phi\|^2\right]$$

- "soft Lipschitz" constraint, non-negative weighting function $f^{c\prime\prime} \geq 0$
- lower variance compared to explicitly adding noise
- **easy to implement & computationally cheap**
- **can train indefinitely without collapse!**

Algorithm 1 Regularized f-GAN. Default values: $\gamma_0 = 2.0$, $\alpha = 0.01$ (with annealing), $\gamma = 0.1$ (without annealing), $n_\psi = 1$

Require: Initial noise variance γ_0, annealing decay factor α, number of discriminator update steps n_ψ per generator iteration, minibatch size m, number of training iterations T
Require: Initial discriminator parameters ω_0, initial generator parameters θ_0
  for $t = 1, ..., T$ **do**
    $\gamma \leftarrow \gamma_0 \cdot \alpha^{t/T}$ # annealing
    for $1, ..., n_\psi$ **do**
      Sample minibatch of real data $\{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(m)}\} \sim \mathbb{P}$.
      Sample minibatch of latent variables from prior $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(m)}\} \sim p(\mathbf{z})$.
      $\omega \leftarrow \omega + \nabla_\omega \left(F(\omega,\theta) - \frac{\gamma}{2}\Omega_f(\omega,\theta) \right)$ # gradient ascent
    end for
    Sample minibatch of latent variables from prior $\{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(m)}\} \sim p(\mathbf{z})$.
    $\theta \leftarrow \theta - \nabla_\theta F(\omega,\theta)$ # gradient descent
  end for

Cross-Testing Protocol

Regularized $\gamma = 0.1$

Pred. \ True Cond.	Pos.	Neg.
Pos.	0.9688	0.0002
Neg.	0.0312	0.9998

Cross-testing: FP: 0.0

Unregularized

Pred. \ True Cond.	Pos.	Neg.
Pos.	1.0	0.0013
Neg.	0.0	0.9987

Cross-testing: FP: 1.0

Cross-Testing:
Classify 10k samples generated by the regularized GAN with the discriminator of the unregularized GAN and vice versa.

=> **Regularized GAN generalizes better!**

Stability across Architectures

RESNET   DCGAN   NO NORMALIZATION   TANH

Sample Quality and Diversity

UNREG.   0.5   ANNEALING   2.0

tf code → github.com/rothk

```
# ---------------------------------------------------------------------------
#     JS-Regularizer
# ---------------------------------------------------------------------------

  def Discriminator_Regularizer(self, D1, D1_logits, D1_arg, D2, D2_logits, D2_arg):
      grad_D1_logits = tf.gradients(D1_logits, D1_arg)[0]
      grad_D2_logits = tf.gradients(D2_logits, D2_arg)[0]
      grad_D1_logits_norm = tf.norm(tf.reshape(grad_D1_logits, [self.batch_size,-1]), axis=1, keep_dims=True)
      grad_D2_logits_norm = tf.norm(tf.reshape(grad_D2_logits, [self.batch_size,-1]), axis=1, keep_dims=True)
      reg_D1 = tf.multiply(tf.square(1.0-D1), tf.square(grad_D1_logits_norm))
      reg_D2 = tf.multiply(tf.square(D2), tf.square(grad_D2_logits_norm))
      return tf.reduce_mean(reg_D1 + reg_D2)
```