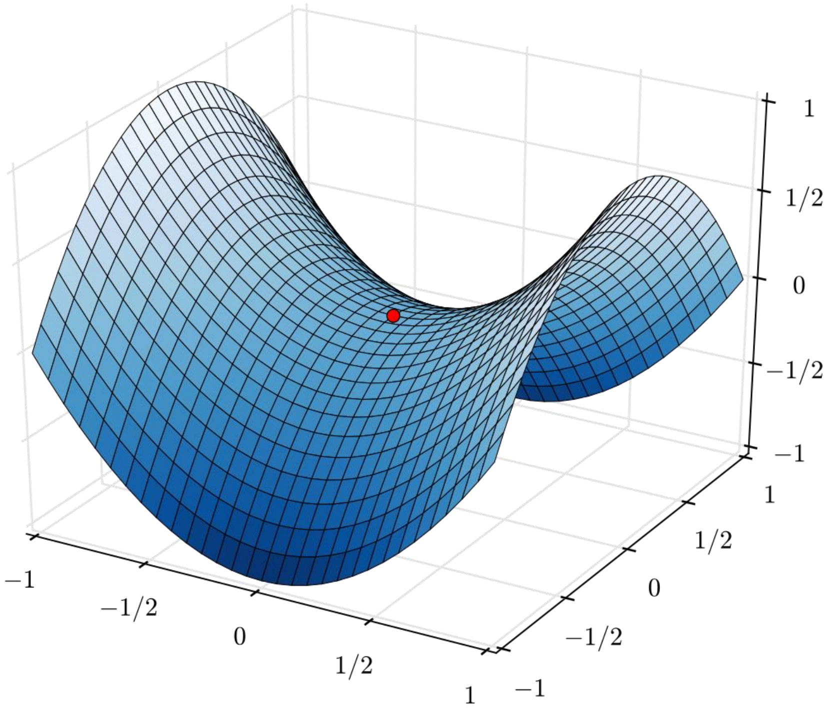
Challenges



Challenge: Vanishing Gradients

- If the discriminator is quite bad
  → the generator gets confused
  → no reasonable generator gradients
- If the discriminator is perfect
  → gradients go to 0, no learning anymore
- Bad if early in the training
  - Easier to train the discriminator than generator

$$J_D = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{data}} \log D(\boldsymbol{x}) \; - \frac{1}{2}\mathbb{E}_z \log(1 - D(G(\boldsymbol{z}))$$

$$J_G = -\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log(D(G(\boldsymbol{z}))$$

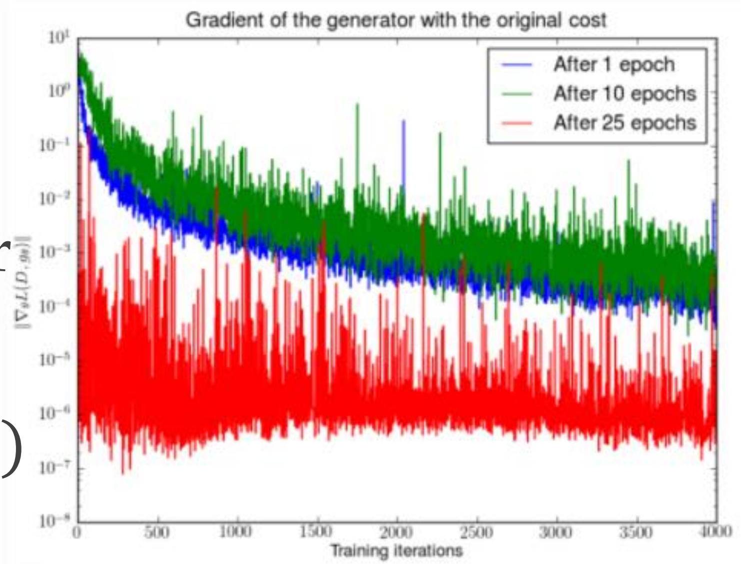
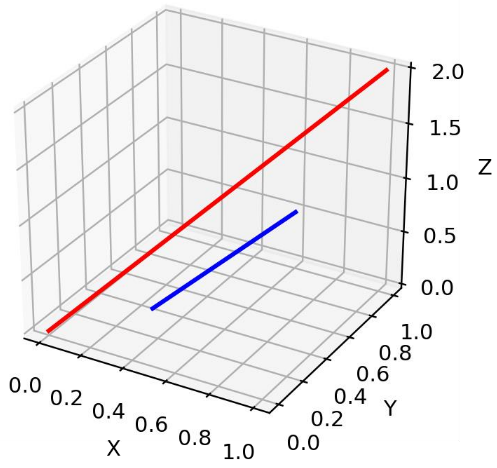
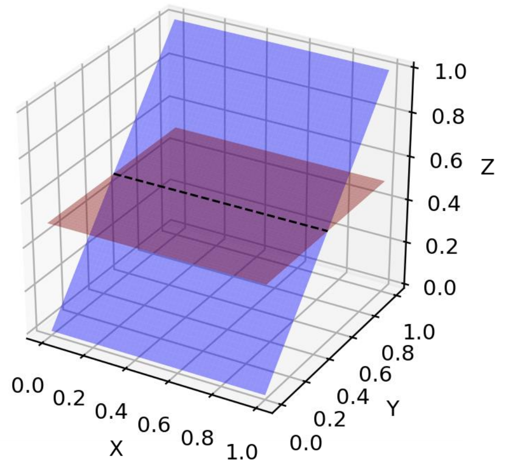


*Fig. 5. First, a DCGAN is trained for 1, 10 and 25 epochs. Then, with the **generator fixed**, a discriminator is trained from scratch and measure the gradients with the original cost function. We see the gradient norms **decay quickly** (in log scale), in the best case 5 orders of magnitude after 4000 discriminator iterations. (Image source: Arjovsky and Bottou, 2017)*

Challenge: Low dimensional supports

- Data lie in low-dim manifolds
- However, the manifold is not known
- During training p_g is not perfect either, especially in the start
- So, the support of p_r and p_g is non-overlapping and disjoint
  → not good for KL/JS divergences
- Easy to find a discriminating line





Challenge: Batch Normalization does not work right away

- Batch-normalization causes strong intra-batch correlation
  - Activations depend on other inputs
  - → Generations depend on other inputs
- Generations look smooth but awkward

Reference batch normalization

- Training with two mini-batches
- Fixed reference mini-batch to compute $\mu_{bn}^{ref}, \sigma_{bn}^{ref}$
- Second mini-batch x_{batch} for training
- Same training, only use $\mu_{bn}^{ref}, \sigma_{bn}^{ref}$ to normalize x_{batch}
- Problem: Overfitting to the reference mini-batch

	Standard mini-batch	Reference mini-batch
Iteration 1	$\frac{dJ^{(1)}}{d\theta}$	μ_{bn}, σ_{bn}
Iteration 2	$\frac{dJ^{(2)}}{d\theta}$	μ_{bn}, σ_{bn}
Iteration 3	$\frac{dJ^{(3)}}{d\theta}$	μ_{bn}, σ_{bn}

Virtual batch normalization

- Append the reference batch to regular mini-batch
- GPU memory is a potential issue

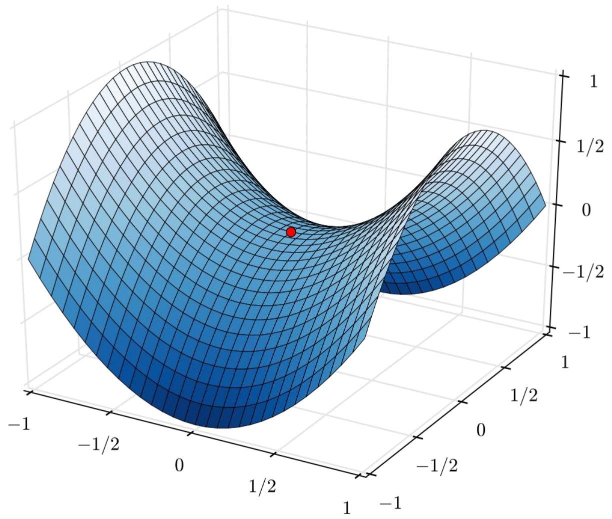
	Standard mini-batch	Reference mini-batch
Iteration 1	$\frac{dJ^{(1)}}{d\theta}$	$\mu_{bn}^{(R)}, \sigma_{bn}^{(R)}$
Iteration 2	$\frac{dJ^{(2)}}{d\theta}$	$\mu_{bn}^{(R)}, \sigma_{bn}^{(R)}$
Iteration 3	$\frac{dJ^{(3)}}{d\theta}$	$\mu_{bn}^{(R)}, \sigma_{bn}^{(R)}$

Balancing generator and discriminator

- Usually the discriminator wins
  - → Good, as the theoretical justification assumes a perfect discriminator
- Usually the discriminator network is bigger than the generator
- Sometimes running discriminator more often than generator works better
  - However, no real consensus
- Do not limit the discriminator to avoid making it too smart
  - Making learning ‘easier’ will not necessarily make generation better
  - Better use non-saturating cost
  - Better use label smoothing

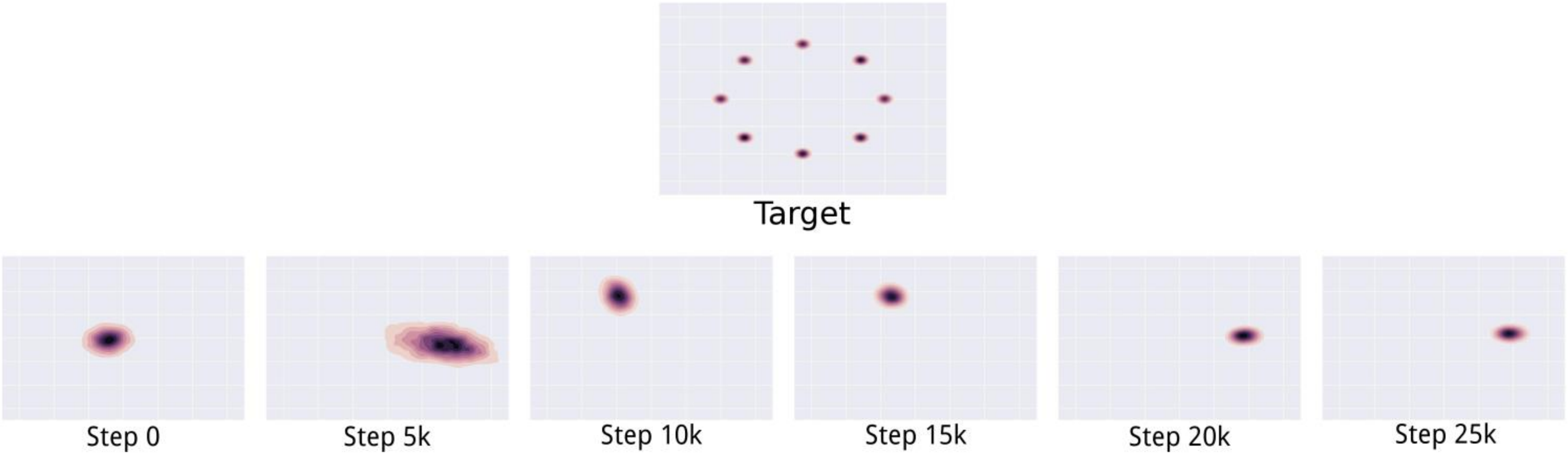
Challenge: Convergence

- Optimization is tricky and unstable
  - finding a saddle point does not imply a global minimum
  - A saddle point is also sensitive to disturbances
- An equilibrium might not even be reached
- Mode-collapse is the most severe form of non-convergence



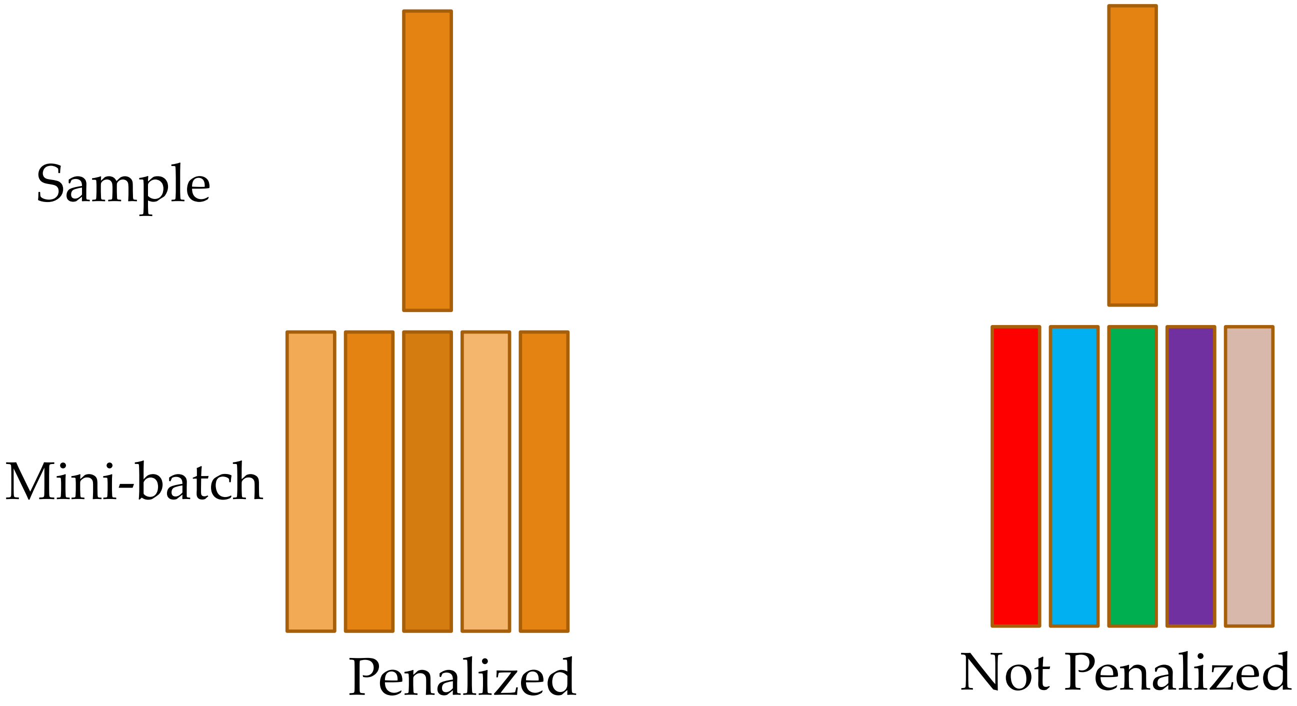
Challenge: mode collapse

- Discriminator converges to the correct distribution
- Generator however places all mass in the most likely point
- All other modes are ignored
  - Underestimating variance
- Low diversity in generating samples



Minibatch features

- Classify each sample by comparing to other examples in the mini-batch
- If samples are too similar, the model is penalized



Challenge: how to evaluate?

- Despite the nice images, who cares?
- It would be nice to quantitatively evaluate the model
- For GANs it is hard to even estimate the likelihood
- In the absence of a precise evaluation metric, do GANs do truly good generations or generations that appeal/fool to the human eye?
  - Can we trust the generations for critical applications, like medical tasks?
  - *'Are humans a good discriminator for the converged generator?'*

Challenge: beyond images

- The generator must be differentiable
- Tasks with discrete outputs (like text) are ruled out
  - modifications are necessary to flow gradients through discrete variables
- Other types of structured data like graphs is also an open problem

A summary of today's open challenges in GANland

- What are the trade-offs between GANs and other generative models?
- What sorts of distributions can GANs model?
- How can we scale GANs beyond image synthesis?
- What can we say about the global convergence of the training dynamics?
- How should we evaluate GANs and when should we use them?
- How does GAN training scale with batch size?
- What is the relationship between GANs and adversarial examples?

https://distill.pub/2019/gan-open-problems/

Feature matching

- Instead of matching image statistics, match feature statistics

$$J_D = \left\| \mathbb{E}_{\boldsymbol{x} \sim p_{data}} f(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})} f\big(G(\boldsymbol{z})\big) \right\|_2^2$$

- f can be any statistic of the data, like the mean or the median

Use labels if possible

- Learning a conditional model $p(\boldsymbol{y}|\boldsymbol{x})$ is often generates better samples
  - Denton et al., 2015
- Even learning $p(\boldsymbol{x}, \boldsymbol{y})$ makes samples look more realistic
  - Salimans et al., 2016
- Conditional GANs are a great addition for learning with labels

One-sided label smoothing

- Default discriminator cost:

```
cross_entropy(1., discriminator(data))
+ cross_entropy(0., discriminator(samples))
```

- One-sided label smoothing:

```
cross_entropy(0.9, discriminator(data))
+ cross_entropy(0., discriminator(samples))
```

- Do not smooth negative labels:

```
cross_entropy(1.-alpha, discriminator(data))
+ cross_entropy(beta, discriminator(samples))
```

Benefits of label smoothing

- Max likelihood often is overconfident
  - Might return accurate prediction, but too high probabilities
- Good regularizer
  - Szegedy et al., 2015
- Does not reduce classification accuracy, only confidence
- Specifically for GANs
  - Prevents discriminator from giving very large gradient signals to generator
  - Prevents extrapolating to encourage extreme samples