IE 534/CS 598 Deep Learning

University of Illinois at Urbana-Champaign

Fall 2018

Lecture 10

The objective function is

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \rho\Big(f(x^i; \theta), y^i \Big). \tag{1}$$

The optimization problem is

$$\min_{\theta} \mathcal{L}(\theta), \tag{2}$$

which is equivalent to

$$\min_{\delta} \mathcal{L}(\theta_0 + \delta). \tag{3}$$

$$
\begin{aligned}
\mathcal{L}(\theta_0 + \delta) &= \sum_{i=1}^{N} \rho\Big(f(x^i; \theta_0 + \delta), y^i \Big) \\
&\approx \sum_{i=1}^{N} \rho\Big(f(x^i; \theta_0) + g_f^i \delta, y^i \Big), \qquad (4)
\end{aligned}
$$

where $g_f^i = \nabla_\theta f(x^i; \theta_0)$.

$$
\begin{aligned}
g &= \nabla_\delta \mathcal{L}(\theta_0 + \delta)\Big|_{\delta=0} \\
&= \sum_{i=1}^{N} \Big(g_f^i \Big)^\top \nabla_z \rho\Big(f(x^i; \theta_0), y^i \Big). \qquad (5)
\end{aligned}
$$

Next, let's calculate the Hessian for $\mathcal{L}(\theta_0 + \delta)$:

$$\begin{aligned} B &= \nabla_{\delta\delta}\mathcal{L}(\theta_0 + \delta)\Big|_{\delta=0} \\ &= \sum_{i=1}^{N} \left(g_f^i\right)^\top \nabla_{zz}\rho\Big(f(x^i;\theta_0), y^i\Big) g_f^i. \end{aligned} \tag{6}$$

Therefore,

$$\mathcal{L}(\theta_0 + \delta) \approx \mathcal{L}(\theta_0) + g^\top \delta + \frac{1}{2}\delta^\top B\delta. \tag{7}$$

Rearranging and solving for first-order optimality,

$$B\delta = -g. \tag{8}$$

- B is positive semi-definite.
- Typically, we use a regularizer

$$B_\lambda = B + \lambda I, \tag{9}$$

  where B_λ is positive definite for $\lambda > 0$.
- Large λ versus small λ
- Adaptive algorithm for selecting λ.

Conjugate gradient method:

- Iterative method for solving $B_\lambda \delta = -g$ when B_λ is positive definite.
- Never explicitly calculate B_λ.
- Never explicitly invert B_λ.
- Instead, use an iteration to solve for δ.

$$\delta_{k+1} = \delta_k + \alpha_k z_k. \tag{10}$$

To evaluate α_k, need to calculate

$$z_k^\top B_\lambda z_k. \tag{11}$$

This can be done without ever explicitly evaluating B_λ!

$$
\begin{aligned}
z^\top B_\lambda z &= z^\top (B + \lambda I) z \\
&= z^\top B z + z^\top \lambda I z \\
&= z^\top B z + \lambda z^\top z. \qquad (12)
\end{aligned}
$$

It remains to evaluate $z^\top B z$.

$$
z^\top B z = \sum_{i=1}^{N} z^\top \left(g_f^i \right)^\top \nabla_{zz} \rho \left(f(x^i; \theta_0), y^i \right) g_f^i z. \qquad (13)
$$

Assume that $\nabla_{zz} \rho \left(z, y \right)$ is positive definite.

$$z^\top B z \;=\; \sum_{i=1}^{N} z^\top \left(g_f^i\right)^\top \nabla_{zz}\rho\Big(f(x^i;\theta_0), y^i\Big) g_f^i z. \qquad (14)$$

- Jacobian-vector product $V^i = g_f^i z$.
- Hessian-vector product $U^i = \nabla_{zz}\rho\Big(f(x^i;\theta_0), y^i\Big) V^i$.
- Jacobian-vector product $T^i = \left(g_f^i\right)^\top U^i$.
- Vector dot product: $z^\top T^i$.

Hessian vector products.

- Compute $V(x) = z^\top \nabla_x f(x)$.
- Compute $\nabla_x V(x)$.
- $\nabla_x V(x) = \big(Hz\big)^\top$.
- Two automatic-differentiation steps!

- Derivation of $\nabla_x V(x) = \big(Hz\big)^\top$.
- Calculate $V(x) = z^\top \nabla_x f(x)$ with Forward AD.
- Calculate $\nabla_x V(x)$ with Reverse AD.

- What would happen if H is not positive semi-definite?
- Our approximation is

$$\mathcal{L}(\theta_0 + \delta) = \mathcal{L}(\theta_0) + g^\top \delta + \frac{1}{2}\delta^\top H \delta. \tag{15}$$

- Conjugate gradient method

Analysis of stochastic gradient descent.

Objective function:

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\theta} \mathbb{E}_V \Big[F(\theta, V) \Big], \tag{16}$$

where $V = (X, Y)$.

Define the projection operator

$$\mathrm{Proj}_{\Theta}(\theta) = \arg\min_{\theta' \in \Theta} \left\| \theta - \theta' \right\|_2 . \tag{17}$$

Projected SGD algorithm:

$$\begin{aligned} \tilde{\theta}_{j+1} &= \theta_j - \gamma_j G(\theta_j, V_j), \\ \theta_{j+1} &= \mathrm{Proj}_{\Theta}(\tilde{\theta}_{j+1}). \end{aligned} \tag{18}$$

where $G(\theta, v) = \nabla_\theta F(\theta, v)$.

Assumptions:

- $\mathcal{L}(\theta)$ is strongly convex.
- Θ is a convex, bounded domain.
- Global minimum θ^* is in the interior of Θ.
- The data V_j is i.i.d.
- $\mathbb{E}\left\|G(\theta_j, V)\right\|^2 \leq M^2$.

Define the distance

$$\begin{aligned} A_j &= \frac{1}{2}\left\|\theta_j - \theta^*\right\|^2, \\ a_j &= \mathbb{E}[A_j]. \end{aligned} \tag{19}$$

We will use the fact that

$$\left\|\mathrm{Proj}_{\Theta}(\theta) - \mathrm{Proj}_{\Theta}(\theta')\right\| \leq \left\|\theta - \theta'\right\|. \tag{20}$$

That is, $\mathrm{Proj}_{\Theta}(\theta)$ is a contraction.

Let the learning rate be

$$\gamma_j = \frac{\alpha}{j}. \tag{21}$$

$$
\begin{aligned}
A_{j+1} &= \frac{1}{2}\left\|\theta_{j+1}-\theta^*\right\|^2 \\
&= \frac{1}{2}\left\|\mathrm{Proj}_{\Theta}(\theta_j-\gamma_j G(\theta_j,V_j),)-\theta^*\right\| \\
&= \frac{1}{2}\left\|\mathrm{Proj}_{\Theta}(\theta_j-\gamma_j G(\theta_j,V_j))-\mathrm{Proj}_{\Theta}(\theta^*)\right\| \\
&\leq \frac{1}{2}\left\|\theta_j-\gamma_j G(\theta_j,V_j)-\theta^*\right\|. \qquad (22)
\end{aligned}
$$

Expanding, we have that

$$
A_{j+1} \leq A_j + \frac{1}{2}\gamma_j^2\left\|G(\theta_j,V_j)\right\|^2 - \gamma_j(\theta_j-\theta^*)^\top G(\theta_j,V_j). \qquad (23)
$$

Taking the expectation,

$$
a_{j+1} \leq a_j + \frac{1}{2}\gamma_j^2\mathbb{E}\big[\left\|G(\theta_j,V_j)\right\|^2\big] - \gamma_j\mathbb{E}\big[(\theta_j-\theta^*)^\top G(\theta_j,V_j)\big]. \qquad (24)
$$

Define

$$\begin{aligned} g(\theta) &:= \nabla_\theta \mathcal{L}(\theta) \\ &= \mathbb{E}[\nabla_\theta F(\theta, V)] \\ &= \mathbb{E}[G(\theta, V)\big|\theta]. \end{aligned} \tag{25}$$

Recall that θ_j and V_j are independent RVs. Using iterated expectations,

$$\begin{aligned} & \mathbb{E}\big[(\theta_j - \theta^*)^\top G(\theta_j, V_j)\big] \\ = \; & \mathbb{E}\Big[\mathbb{E}\big[(\theta_j - \theta^*)^\top G(\theta_j, V_j)\big|\theta_j\big]\Big] \\ = \; & \mathbb{E}\Big[\big[(\theta_j - \theta^*)^\top \mathbb{E}\big[G(\theta_j, V_j)\big|\theta_j\big]\Big] \\ = \; & \mathbb{E}\Big[\big[(\theta_j - \theta^*)^\top g(\theta_j)\Big]. \end{aligned} \tag{26}$$

Therefore, we have the inequality

$$a_{j+1} \leq a_j + \frac{1}{2}\gamma_j^2 \mathbb{E}\left[\, \|G(\theta_j, V_j)\|^2 \,\right] - \gamma_j \mathbb{E}\Big[\big[(\theta_j - \theta^*)^\top g(\theta_j)\Big]. \quad (27)$$

Let's first address the last term on the RHS. Recall that a strongly convex function $f(\theta)$ is

$$f(\theta') \geq f(\theta) + (\theta' - \theta)^\top \nabla f(\theta) + \frac{1}{2} c \left\|\theta - \theta'\right\|_2^2, \quad (28)$$

for any θ', θ. Rearranging, we have that

$$-(\theta' - \theta)^\top \nabla f(\theta)) \quad \geq \quad f(\theta) - f(\theta') + \frac{1}{2} c \left\|\theta - \theta'\right\|_2^2. \quad (29)$$

$$-(\theta' - \theta)^\top \nabla f(\theta) \geq f(\theta) - f(\theta') + \frac{1}{2} c \left\| \theta - \theta' \right\|_2^2 . \tag{30}$$

Let $\theta' = \theta^*$. Then,

$$\begin{aligned} -(\theta^* - \theta)^\top \nabla f(\theta) &\geq f(\theta) - f(\theta^*) + \frac{1}{2} c \left\| \theta - \theta^* \right\|_2^2 \\ &\geq \frac{1}{2} c \left\| \theta - \theta^* \right\|_2^2 . \end{aligned} \tag{31}$$

Therefore,

$$(\theta - \theta^*)^\top \nabla f(\theta) \geq \frac{1}{2} c \left\| \theta - \theta^* \right\|_2^2 . \tag{32}$$

$$(\theta - \theta^*)^\top \nabla f(\theta) \geq \frac{1}{2} c \, \|\theta - \theta^*\|_2^2 \,. \tag{33}$$

This implies that

$$\begin{aligned} \mathbb{E}\Big[\big[(\theta_j - \theta^*)^\top g(\theta_j)\Big] &\geq \mathbb{E}\Big[\frac{1}{2} c \, \|\theta_j - \theta^*\|_2^2\Big] \\ &\geq c a_j. \end{aligned} \tag{34}$$

Therefore,

$$\begin{aligned} a_{j+1} &\leq a_j + \frac{1}{2}\gamma_j^2 \mathbb{E}\big[\, \|G(\theta_j, V_j)\|^2 \,\big] - \gamma_j c a_j \\ &\leq (1 - 2c\gamma_j) a_j + \frac{1}{2}\gamma_j^2 M^2. \end{aligned} \tag{35}$$

Recall that $\gamma_j = \frac{\alpha}{j}$, which leads to

$$a_{j+1} \leq \left(1 - \frac{2c\alpha}{j}\right) a_j + \frac{1}{2}\frac{\alpha^2 M^2}{j^2}. \tag{36}$$

Proof by Induction.

Let $\tilde{M} = \max(M, \|\theta_0\|)$. Then,

$$a_{j+1} \leq \left(1 - \frac{2c\alpha}{j}\right) a_j + \frac{1}{2}\frac{\alpha^2 \tilde{M}^2}{j^2}. \tag{37}$$

Assume that $a_j = \frac{Q}{j}$ where $Q = \frac{1}{2}\frac{\alpha^2 \tilde{M}^2}{2c\alpha - 1}$. Select α such that $2c\alpha - 1 > 0$.

Then,

$$\begin{aligned} a_{j+1} &\leq \frac{1}{2}\frac{\alpha^2 \tilde{M}^2}{2c\alpha - 1}\left[\frac{j - 2c\alpha}{j^2} + \frac{2c\alpha - 1}{j^2}\right] \\ &= \frac{1}{2}\frac{\alpha^2 \tilde{M}^2}{2c\alpha - 1}\frac{j - 1}{j^2}. \end{aligned} \tag{38}$$

$$a_{j+1} \leq \frac{1}{2}\frac{\alpha^2\tilde{M}^2}{2c\alpha-1}\frac{j-1}{j^2}. \tag{39}$$

Note that

$$(j+1)(j-1) = j^2 - 1 \leq j^2. \tag{40}$$

Therefore,

$$\begin{aligned} a_{j+1} &\leq \frac{1}{2}\frac{\alpha^2\tilde{M}^2}{2c\alpha-1}\frac{j-1}{(j+1)(j-1)} \\ &\leq \frac{1}{2}\frac{\alpha^2\tilde{M}^2}{2c\alpha-1}\frac{1}{j+1} \\ &\leq \frac{Q}{j+1}. \end{aligned} \tag{41}$$

Proof for convergence rate heavily relies upon:

- Selecting a sufficiently large learning rate magnitude

$$2c\alpha - 1 > 0, \tag{42}$$

  where $\gamma_j = \frac{\alpha}{j}$.

- Objective function $\mathcal{L}(\theta)$ being strongly convex
- Projection onto the compact set Θ.

Practical challenge: we don't a priori know c, so difficult to select the optimal α.

If we did not have the projection onto the compact set Θ, it would not be appropriate to assume that

$$\mathbb{E}\,\|G(\theta_j, V)\|^2 \leq M^2. \tag{43}$$

θ_j may grow very large since it is not restricted to the bounded set Θ!

- Suppose that $\mathcal{L}(\theta)$ is in general non-convex, but is strongly convex in a neighborhood $\|\theta - \theta^*\| < \delta$.
- Non-zero probability that θ_j will escape the neighborhood, and then the analysis does not work.

Convergence analysis of stochastic gradient descent in non-convex case.

Objective function:

$$\mathcal{L}(\theta) = \mathbb{E}_{(X,Y)}\big[\rho(f(X;\theta), Y)\big]. \tag{44}$$

SGD algorithm:

$$\theta^{(\ell+1)} = \theta^{(\ell)} - \alpha^{(\ell)}\nabla_\theta \rho(f(x^{(\ell)};\theta^{(\ell)}), y^{(\ell)}). \tag{45}$$

The parameter at the $(\ell+1)$-th iteration can be re-written as:

$$\begin{aligned}
\theta^{(\ell+1)} &= \theta^{(\ell)} - \alpha^{(\ell)}\mathbb{E}_{X,Y}\big[\nabla_\theta \rho(f(X;\theta^{(\ell)}), Y)\big] \\
&+ \alpha^{(\ell)}\Big(\mathbb{E}_{X,Y}\big[\nabla_\theta \rho(f(X;\theta^{(\ell)}), Y)\big] - \nabla_\theta \rho(f(x^{(\ell)};\theta^{(\ell)}), y^{(\ell)})\Big) \\
&= \theta^{(\ell)} - \alpha^{(\ell)}\nabla_\theta \mathcal{L}(\theta^{(\ell)}) \\
&+ \alpha^{(\ell)}\Big(\mathbb{E}_{X,Y}\big[\nabla_\theta \rho(f(X;\theta^{(\ell)}), Y)\big] - \nabla_\theta \rho(f(x^{(\ell)};\theta^{(\ell)}), y^{(\ell)})\Big).
\end{aligned} \tag{46}$$

- In order for the stochastic gradient descent algorithm to converge to a stationary point of the objective function $\mathcal{L}(\theta)$, its dynamics must asymptotically follow the descent direction $\nabla_\theta \mathcal{L}(\theta^{(\ell)})$.

- In order for this to occur, the "contribution" from the noise term

$$N^{(\ell)} = \alpha^{(\ell)} \left(\mathbb{E}_{X,Y}\left[\nabla_\theta \rho(f(X;\theta^{(\ell)}), Y)\right] - \nabla_\theta \rho(f(x^{(\ell)};\theta^{(\ell)}), y^{(\ell)}) \right)$$

  must vanish as $\ell \to \infty$.

- It turns out that these contributions from the noise term will disappear if $\alpha^{(\ell)}$ appropriately decays as $\ell \to \infty$.

Assume $\rho(f(x;\theta), y) \in C_b^2$.[1] For $\ell > L$, (46) can be re-written via a telescoping series and a Taylor expansion as

$$\begin{aligned}
\mathcal{L}(\theta^{(\ell)}) - \mathcal{L}(\theta^{(L)}) &= -\sum_{m=L}^{\ell-1} \alpha^{(m)} \left(\nabla_\theta \mathcal{L}(\theta^{(m)}) \right)^\top \nabla_\theta \mathcal{L}(\theta^{(m)}) \\
&+ \sum_{m=L}^{\ell-1} \left(\nabla_\theta \mathcal{L}(\theta^{(m)}) \right)^\top N^{(m)} \\
&+ \sum_{m=L}^{\ell-1} \left[\left(\alpha^{(m)}\right)^2 \times \text{Higher order terms} \right] \quad (47)
\end{aligned}$$

[1] C_b^k is the space functions with bounded and continuous derivatives up to order k.

Recall that the learning rate must satisfy

$$\sum_{m=0}^{\infty} \left(\alpha^{(m)}\right)^2 < C, \tag{48}$$

for some finite constant C.
Then, for any $\epsilon > 0$, there exists an L such that

$$\sum_{m=L}^{\infty} \left(\alpha^{(m)}\right)^2 < \epsilon. \tag{49}$$

Since $\rho(f(x;\theta), y) \in C_b^2$, the higher order terms are bounded.
Then, for any any $\epsilon > 0$, there exists an L such that for all $\ell > L$

$$\left| \sum_{m=L}^{\ell-1} \left[\left(\alpha^{(m)}\right)^2 \times \text{Higher order terms}\right] \right| < K \sum_{m=L}^{\ell-1} \left(\alpha^{(m)}\right)^2 < \epsilon. \tag{50}$$

We also can show that, as $L \to \infty$,

$$\left| \sum_{m=L}^{\infty} \left(\nabla_\theta \mathcal{L}(\theta^{(m)}) \right)^\top N^{(m)} \right| \overset{a.s.}{\to} 0. \tag{51}$$

(Proof omitted.)

- Consequently, the contributions of the noise term and higher-order terms in equation (47) become small as L become large.
- The dominating term for large L is the first term in (47).
- That is, for large L, we have that

$$\mathcal{L}(\theta^{(\ell)}) - \mathcal{L}(\theta^{(L)}) \approx -\sum_{m=L}^{\ell-1} \alpha^{(m)} \left(\nabla_\theta \mathcal{L}(\theta^{(m)}) \right)^\top \nabla_\theta \mathcal{L}(\theta^{(m)}). \quad (52)$$

- This is a sum of non-negative terms, which implies that $\mathcal{L}(\theta^{(\ell)}) - \mathcal{L}(\theta^{(L)}) < 0$, i.e. it decreases the objective function.

$$\mathcal{L}(\theta^{(\ell)}) - \mathcal{L}(\theta^{(L)}) \approx -\sum_{m=L}^{\ell-1} \alpha^{(m)} \left(\nabla_\theta \mathcal{L}(\theta^{(m)})\right)^\top \nabla_\theta \mathcal{L}(\theta^{(m)}). \quad (53)$$

- We can easily see that the algorithm should converge to a stationary point $\nabla_\theta \mathcal{L}(\theta) = 0$. Here's a heuristic argument.

- If $\left(\nabla_\theta \mathcal{L}(\theta^{(m)})\right)^\top \nabla_\theta \mathcal{L}(\theta^{(m)}) \geq \delta > 0$, then $\mathcal{L}(\theta^{(\ell)}) - \mathcal{L}(\theta^{(L)}) \leq -\delta \sum_{m=L}^{\ell-1} \alpha^{(m)}$.

- Since $\sum_{m=L}^{\ell-1} \alpha^{(m)} = \infty$, $\mathcal{L}(\theta^{(\ell)}) \to -\infty$.

- However, since $\mathcal{L}(\theta)$ is bounded, this cannot be true, and therefore there must be some $m > L$ such that $\left(\nabla_\theta \mathcal{L}(\theta^{(m)})\right)^\top \nabla_\theta \mathcal{L}(\theta^{(m)}) < \delta$.

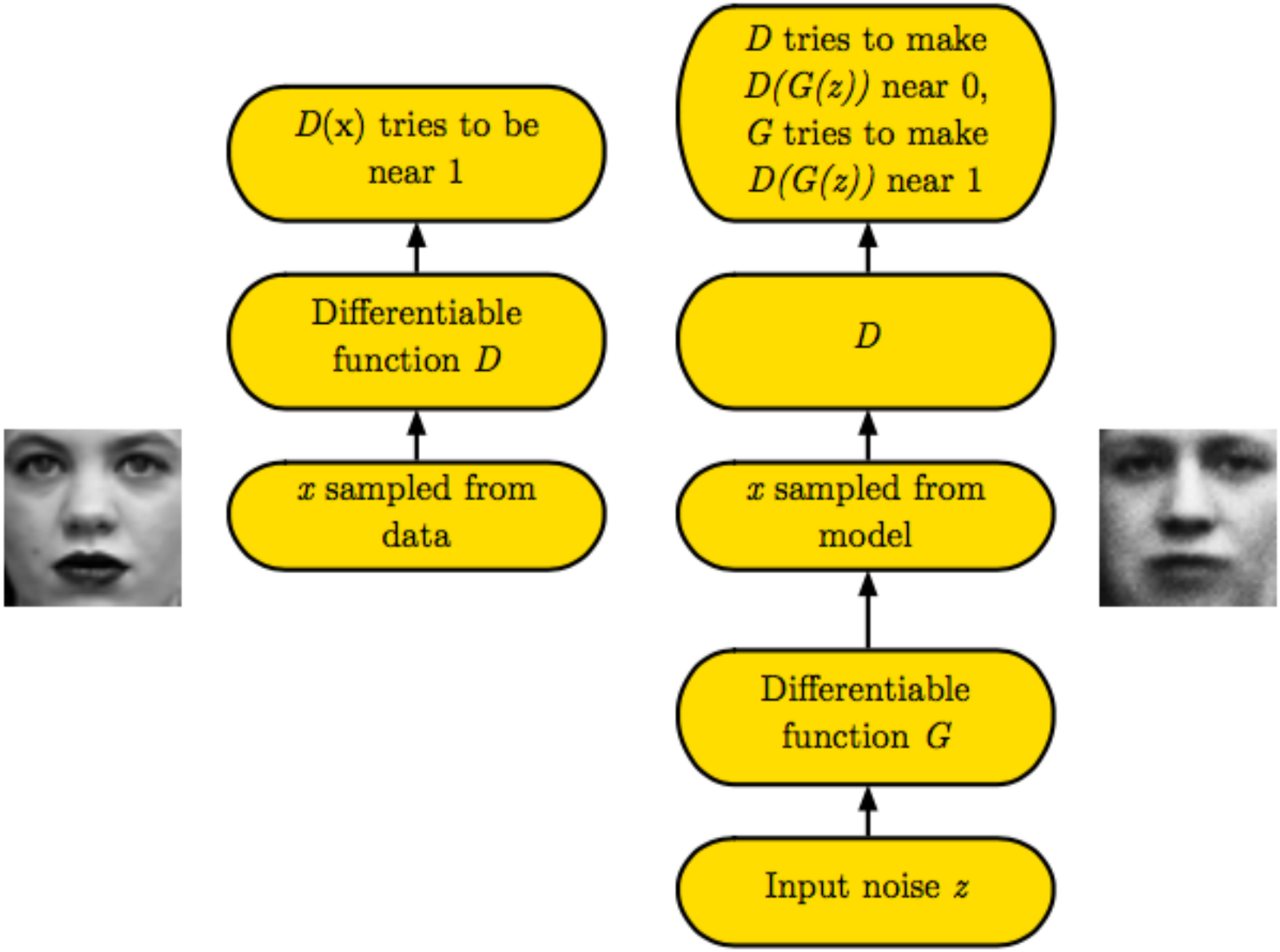
Generative Adversarial Networks.

$$J(\theta^D, \theta^G) = \mathbb{E}_X\left[\log f_D(X;\theta^D)\right] + \mathbb{E}_Z\left[\log\left(1 - f_D\big(f_G(Z;\theta^G)\big), \theta^D\right)\right],$$

where X is the distribution of the "real data" and Z is a random variable (e.g., a Gaussian).

The optimization problem is

$$\min_{\theta^G} \max_{\theta^D} J(\theta^D, \theta^G). \tag{54}$$



Source: NIPS 2016 Tutorial: Generative Adversarial Networks by I.

Applications of generative models:

- Data simulation (similar in spirit to data augmentation)
- Semi-supervised learning ($K + 1$ classes), which can use unlabeled data.
- Analyze robustness of a predictive model
- Image super-resolution

Stochastic gradient descent algorithm:

- For k steps:
  - Update the Discriminator:

$$\begin{aligned}\theta^D &= \theta^D + \alpha g^D, \\ g^D &= \nabla_{\theta^D} \frac{1}{M} \sum_{i=1}^{M} \Big[\log f_D(x^i; \theta^D) + \log \big(1 - f_D\big(f_G(z^i; \theta^G)\big), \theta^D\big) \Big].\end{aligned}$$

  - Update the Generator:

$$\begin{aligned}\theta^G &= \theta^G + \alpha g^G, \\ g^G &= \nabla_{\theta^G} \frac{1}{M} \sum_{i=1}^{M} \log \big(1 - f_D\big(f_G(z^i; \theta^G)\big), \theta^D\big).\end{aligned}$$

Mode collapse.

- "Mode collapse" is when the generator produces the same output for many different inputs Z.
- "Partial mode collapse": different but very similar outputs for different inputs Z.
- Challenges with larger-scale images.

Figure 22: An illustration of the mode collapse problem on a two-dimensional toy dataset. In the top row, we see the target distribution p_{data} that the model should learn. It is a mixture of Gaussians in a two-dimensional space. In the lower row, we see a series of different distributions learned over time as the GAN is trained. Rather than converging to a distribution containing all of the modes in the training set, the generator only ever produces a single mode at a time, cycling between different modes as the discriminator learns to reject each one. Images from Metz *et al.* (2016).

Source: NIPS 2016 Tutorial: Generative Adversarial Networks by I. Goodfellow

Source: NIPS 2016 Tutorial: Generative Adversarial Networks by I. Goodfellow

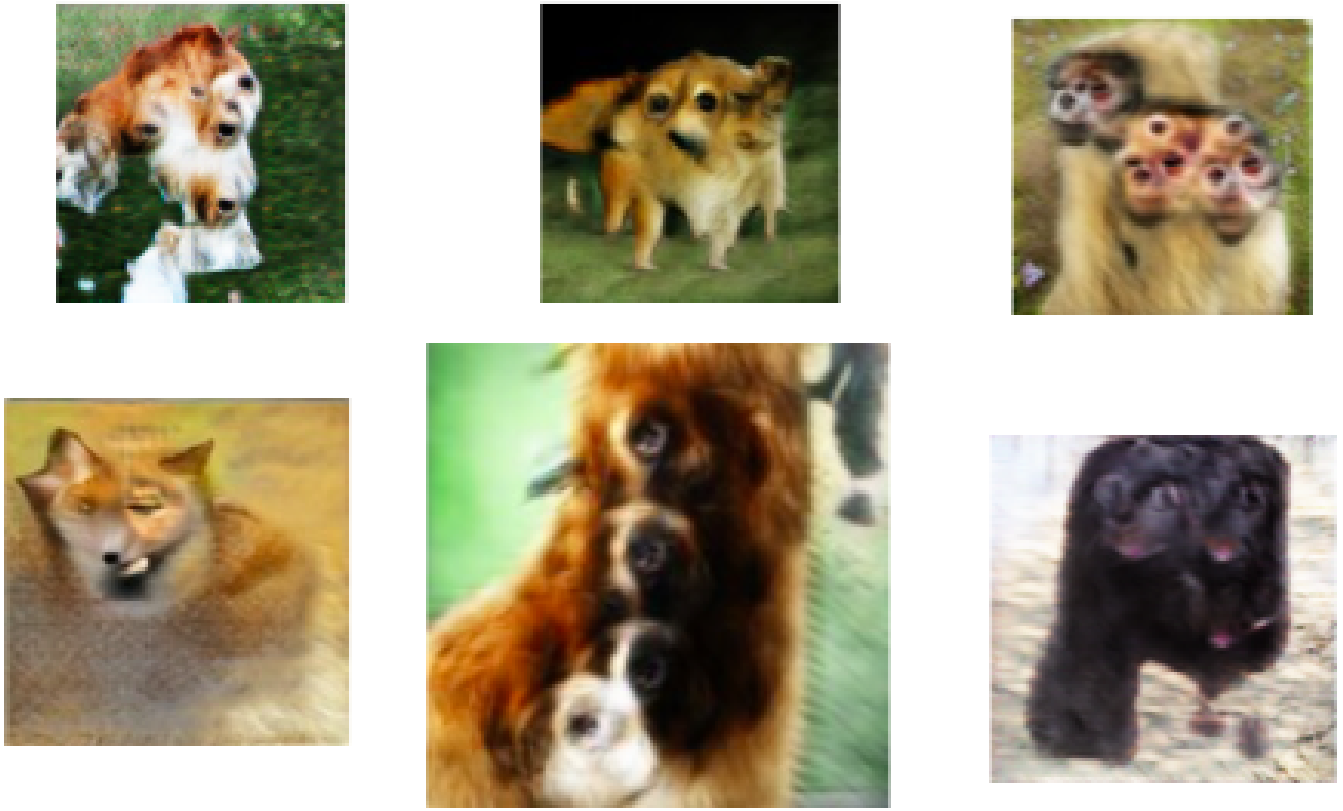


Figure 29: GANs on 128 × 128 ImageNet seem to have trouble with counting, often generating animals with the wrong number of body parts.

Source: NIPS 2016 Tutorial: Generative Adversarial Networks by I. Goodfellow

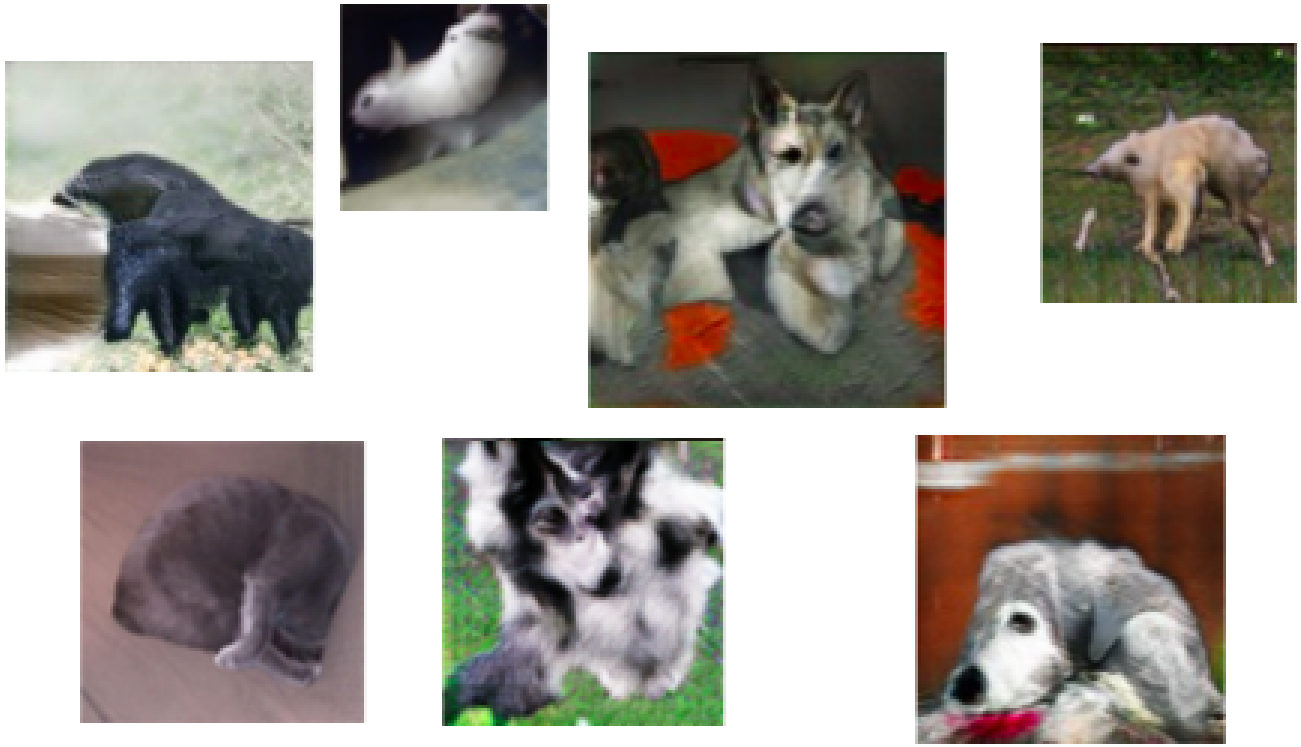


Figure 30: GANs on 128×128 ImageNet seem to have trouble with the idea of three-dimensional perspective, often generating images of objects that are too flat or highly axis-aligned. As a test of the reader's discriminator network, one of these images is actually real.

Source: NIPS 2016 Tutorial: Generative Adversarial Networks by I. Goodfellow

Wasserstein-GAN.

The Wasserstein distance between two probability measures μ and ν is

$$W(\mu, \nu) = \inf_{\gamma \in \prod(\mu,\nu)} \mathbb{E}_{X,G}\Big[\, \|X - G\|_1 \,\Big], \tag{55}$$

where $\prod(\mu, \nu)$ is any probability measure with marginals μ and ν.

Some properties of the Wasserstein distance:

- Let $G \sim \nu$ and $X \sim \mu$.
- If $W(\mu, \nu) = 0$, then $\mu = \nu$.

Some properties of the Wasserstein distance:

- Let $G^N \sim \nu^N$ and $X \sim \mu$.
- If $W(\mu, \nu^N) \to 0$ as $N \to \infty$, then $G^N \xrightarrow{d} X$.
- $G^N \xrightarrow{d} X$ denotes "convergence in distribution", i.e.

$$\mathbb{E}\big[f(G^N)\big] \to \mathbb{E}\big[f(X)\big], \tag{56}$$

  for any continuous, bounded function f.

By the Kantorovich-Rubinstein duality,

$$W(\mu, \nu) = \sup_{\|f\|_L \leq 1} \left[\mathbb{E}_{X \sim \mu}\big[f(X)\big] - \mathbb{E}_{G \sim \nu}\big[f(G)\big] \right], \tag{57}$$

where $\{f : \|f\|_L \leq 1\}$ is the set of functions f such that

$$\left\| f(x) - f(x') \right\|_1 \leq \left\| x - x' \right\|_1 . \tag{58}$$

That is, functions f which are Lipschitz continuous with Lipschitz constant $K = 1$.

In the context of GAN, let $g_\theta(z)$ be the generator network and set

$$G = g_\theta(Z). \tag{59}$$

Let μ be the distribution of X and let ν_θ be the distribution of G. Then, our objective function is

$$\begin{aligned} W(\mu, \nu_\theta) &= \sup_{\|f\|_L \leq 1} \Big[\mathbb{E}_{X\sim\mu}\big[f(X)\big] - \mathbb{E}_{G\sim\nu}\big[f(G)\big] \Big] \\ &= \sup_{\|f\|_L \leq 1} \Big[\mathbb{E}_{X\sim\mu}\big[f(X)\big] - \mathbb{E}_{Z\sim p(z)}\big[f\big(g_\theta(Z)\big)\big] \Big] \end{aligned} \tag{60}$$

Of course, this is not numerically tractable since the supremum is over $\{f : \|f\|_L \leq 1\}$.

$$W(\mu, \nu_\theta) = \sup_{\|f\|_L \leq 1} \left[\mathbb{E}_{X \sim \mu}\big[f(X)\big] - \mathbb{E}_{Z \sim p(z)}\big[f\big(g_\theta(Z)\big)\big] \right]. \quad (61)$$

We want to minimize over ν_θ, which leads to the optimization problem

$$\min_\theta W(\mu, \nu_\theta) = \min_\theta \sup_{\|f\|_L \leq 1} \left[\mathbb{E}_{X \sim \mu}\big[f(X)\big] - \mathbb{E}_{Z \sim p(z)}\big[f\big(g_\theta(Z)\big)\big] \right]. (62)$$

Let

$$f^* = \arg \max_{\|f\|_L \leq 1} \Big[\mathbb{E}_{X \sim \mu} \big[f(X) \big] - \mathbb{E}_{Z \sim p(z)} \big[f\big(g_\theta(Z)\big) \big] \Big]. \tag{63}$$

Then, we could take a gradient descent step:

$$\begin{aligned} \nabla_\theta W(\mu, \nu_\theta) &= \nabla_\theta \mathbb{E}_{Z \sim p(z)} \Big[f^*\big(g_\theta(Z)\big) \Big], \\ &= \mathbb{E}_{Z \sim p(z)} \big[\nabla_\theta f^* \Big(g_\theta(Z)\big) \Big]. \end{aligned} \tag{64}$$

In fact, we could take a stochastic gradient descent step:

$$\theta = \theta + \alpha \nabla_\theta f^*\big(g_\theta(Z)\big). \tag{65}$$

How to numerically estimate f^*?

- Instead of optimizing over the set of functions $\{f : \|f\|_L \leq 1\}$, let's optimize over the class of neural networks $f_w(x)$.

- The objective function becomes

$$\sup_w \left[\mathbb{E}_{X \sim \mu}\big[f_w(X)\big] - \mathbb{E}_{Z \sim p(z)}\big[f_w\big(g_\theta(Z)\big)\big] \right]. \tag{66}$$

Define

$$
\begin{aligned}
J(\mu, \nu_\theta) &:= \sup_{\|f\|_L \le K} \mathbb{E}_{X\sim\mu}\big[f(X)\big] - \mathbb{E}_{Z\sim p(z)}\big[f\big(g_\theta(Z)\big)\big] \\
&= K \sup_{\|f\|_L \le K} \mathbb{E}_{X\sim\mu}\big[\frac{1}{K} f(X)\big] - \mathbb{E}_{Z\sim p(z)}\big[\frac{1}{K} f\big(g_\theta(Z)\big)\big] \\
&= K \sup_{h=\frac{1}{K}f, \|f\|_L \le K} \mathbb{E}_{X\sim\mu}\big[h(X)\big] - \mathbb{E}_{Z\sim p(z)}\big[h\big(g_\theta(Z)\big)\big] \\
&= K \sup_{\|h\|_L \le 1} \mathbb{E}_{X\sim\mu}\big[h(X)\big] - \mathbb{E}_{Z\sim p(z)}\big[h\big(g_\theta(Z)\big)\big] \\
&= KW(\mu, \nu_\theta), \qquad (67)
\end{aligned}
$$

since

$$
\begin{aligned}
\big\|h(x) - h(x')\big\| &= \frac{1}{K}\big\|f(x) - f(x')\big\| \\
&\le \frac{1}{K} \cdot K \big\|x - x'\big\| \\
&= \big\|x - x'\big\|. \qquad (68)
\end{aligned}
$$

Therefore, minimizing $J(\mu, \nu_\theta)$ is equivalent to minimizing $W(\mu, \nu_\theta)$. That is,

$$\nabla_\theta W(\mu, \nu_\theta) = \frac{1}{K} \nabla_\theta J(\mu, \nu_\theta). \tag{69}$$

How to numerically estimate f^*?

- Instead of optimizing over the set of functions $\{f : \|f\|_L \leq K\}$, let's optimize over the class of neural networks $\{f_w(x) : w \in \mathcal{W}\}$.
- Neural networks can approximate any continuous function on a compact set
- Therefore, $\{f_w(x) : w \in \mathcal{W}\}$ is a good approximation for $\{f : \|f\|_L \leq K\}$!
- The objective function is approximated as

$$J(\mu, \nu_\theta) \approx \sup_{w \in \mathcal{W}} \mathbb{E}_{X \sim \mu}\big[f_w(X)\big] - \mathbb{E}_{Z \sim p(z)}\big[f_w\big(g_\theta(Z)\big)\big].$$

Find w^* with gradient descent:

$$\begin{aligned} \tilde{w} &= w + \alpha \nabla_w \Big[\mathbb{E}_{X \sim \mu} \big[f_w(X) \big] - \mathbb{E}_{Z \sim p(z)} \big[f_w \big(g_\theta(Z) \big) \big] \Big], \\ w &= \max(\min(\tilde{w}, c), -c). \end{aligned} \tag{70}$$

Find w^* with stochastic gradient descent:

$$\begin{aligned} \tilde{w} &= w + \alpha \Big(\nabla_w f_w(X) - \nabla_w f_w \big(g_\theta(Z) \big) \Big), \\ w &= \max(\min(\tilde{w}, c), -c). \end{aligned} \tag{71}$$

Collecting our results, the WGAN algorithm is:

- Repeat until convergence:
  - Sample (X, Z)
  - Take a step:

$$\begin{aligned} \tilde{w} &= w + \alpha\Big(\nabla_w f_w(X) - \nabla_w f_w\big(g_\theta(Z)\big)\Big), \\ w &= \max(\min(\tilde{w}, c), -c). \end{aligned} \tag{72}$$

- Samples Z
- Take a step:

$$\theta = \theta + \alpha \nabla_\theta f_w\big(g_\theta(Z)\big). \tag{73}$$

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of priors.
    $g_w \leftarrow \nabla_w [\frac{1}{m} \sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

Source: Wasserstein Generative Adversarial Networks by Arjovsky, Chintala, and Bottou.

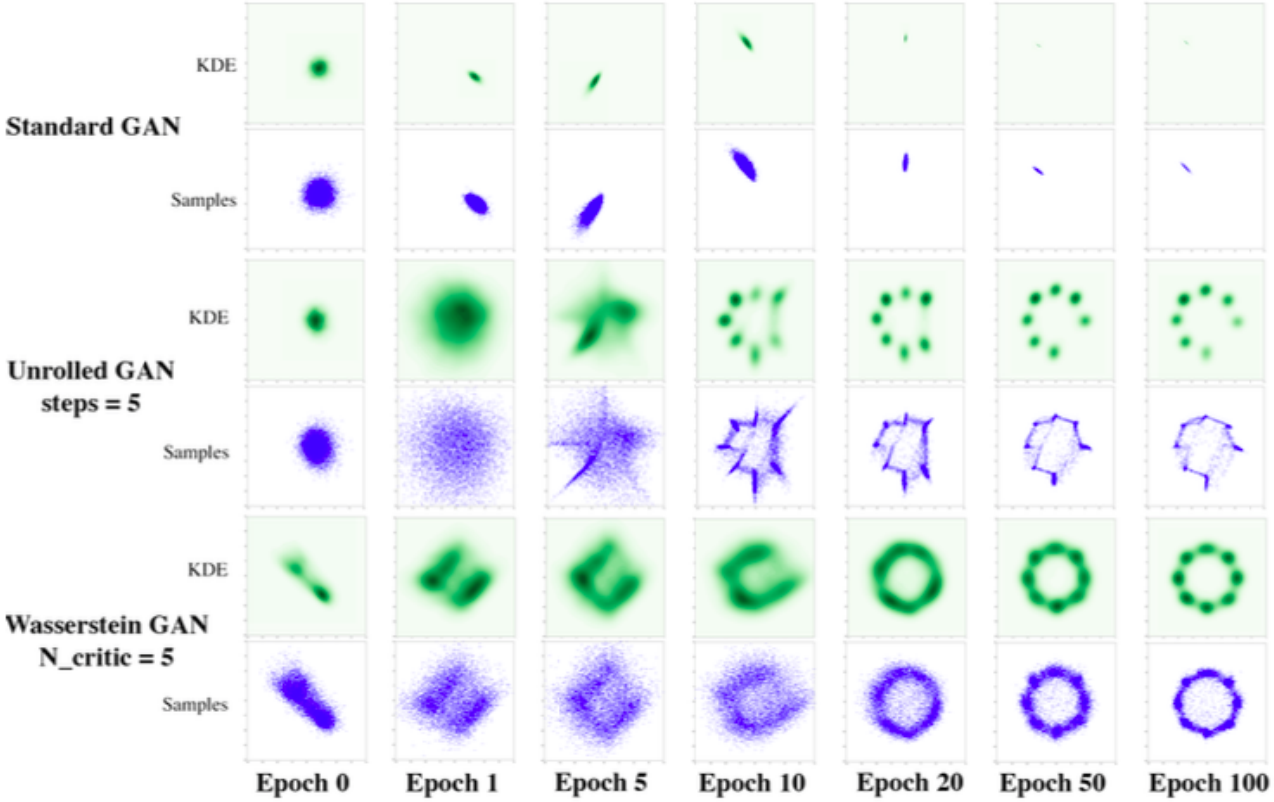


Figure 2: Different methods learning a mixture of 8 gaussians spread in a circle. WGAN is able to learn the distribution without mode collapse. An interesting fact is that the WGAN (much like the Wasserstein distance) seems to capture first the low dimensional structure of the data (the approximate circle) before matching the specific bumps in the density. Green: KDE plots. Blue: samples from the model.

Source: Wasserstein Generative Adversarial Networks by Arjovsky, Chintala, and Bottou.