# WGAN Loss

$$\begin{cases} L = \sum_{x \in \text{real}} h(x) - \sum_{z \in \text{latent}} h(G(z)) \\ \min_{G} \max_{h \in \text{Lipschitz}} L \end{cases}$$

Enforcing Lipschitz: many approaches

- original paper: "weight clipping"

after each weight update

$w \to \text{clip}(w, -c, c)$ $\begin{pmatrix} \text{if } w > c & w = c \\ \text{if } w < -c & w = -c \end{pmatrix}$

This enforces K-Lipschitz crudely

roughly: NN output $h = g_1 \circ W_1 \circ g_2 \circ W_2 \circ \cdots \circ g_L \circ W_L(x)$

so $|\nabla_x h| \sim |g_1' W_1 g_2' W_2 \cdots g_L' W_L|$

so if weights bounded, $|\nabla_x h|$ bounded ✓
& activation gradients bounded } then

- Better way: gradient penalty

add regularizer to loss $\delta L = \lambda \sum_{\hat{x} \sim p} \left( |\nabla_x h(\hat{x})|^2 - 1 \right)^2$

might not be state of the art...

$p$ : dist'n of pts $\hat{x} = tx + (1-t)y$
$x, y \sim$ data, gen
$t \sim U(0,1)$

Some theory behind this...

not clear why seems to require $|\nabla_x h| = 1$
instead of $\leq 1$ ... but empirically works

- WGAN - MNIST example
- ATLAS Fast calogan example
  - Michele's slides

---

## Next generative model framework: Variational autoencoders

- example of a latent variable model

data $x \longrightarrow$ latent variable $z$
encodes "meaning" of $x$ (aka embedding)
ideally in a much reduced, simpler space

To sample:

$z \sim p(z)$ ← "prior" some fixed, simple dist'n

$x \sim p_\theta(x|z)$ Thus gives $(x, z) \sim p(x, z)$

Throw away $z \rightarrow x \sim p(x)$.

Objective: max likelihood WANT: $p(x) = p_{data}(x)$.

$$\max_\theta \sum_{x \sim p_{data}} \log p_\theta(x) = \max_\theta \sum_{x \sim data} \log \int p_\theta(x|z) p(z) dz$$

↓ direct eval via MC integ?

$$\max_\theta \sum_{x \sim data} \log \frac{1}{N} \sum_{z \sim p(z)} p_\theta(x|z)$$

- Computationally expensive
- Noisy gradients
- Curse of dimensionality (MC needs large N in higher dim)

$\rightarrow$ direct MC generally fails!

Instead: variational approach + tractable lower bound on likelihood

introduce $q_\phi(z|x)$ $\leftarrow$ proposal dist'n for more efficient sampling than prior

$$\log p(x) = \log \int dz\, p_\theta(x|z)\, p(z)\, \frac{q_\phi(z|x)}{q_\phi(z|x)}$$

$$\approx \log \frac{1}{N} \sum_{z \sim q_\phi(z|x)} \frac{p_\theta(x|z)\, p(z)}{q_\phi(z|x)}$$

$\rightarrow$ sampling would be most efficient if $q_\phi(z|x) \sim p_\theta(x|z)\, p(z)$

i.e. integrand $\mathcal{O}(1)$ always

"Jensen ineq"

$\log \frac{1}{N} \sum f \geq \frac{1}{N} \sum \log f$

(AKA AM-GM ineq)

$$\geq \frac{1}{N} \sum_{z \sim q_\phi(z|x)} \log \frac{p_\theta(x|z)\, p(z)}{q_\phi(z|x)}$$

"evidence lower bound"

ELBO

thing inside log wants to be $p(x)$

Bayesian evidence

So $q_\phi$ wants to be posterior!

more to come...

The ELBO is tractable!

$$\max_{\theta,\phi} \sum_{x \sim p_{data}(x)} \text{ELBO} = \max_{\theta,\phi} \sum_{\substack{x \sim p_{data}(x) \\ z \sim q_\phi(z|x)}} \log \frac{p_\theta(x|z)\, p(z)}{q_\phi(z|x)}$$

Just train w/ batches of $x$ and $z$ together