Structured Model Learning
Generative Adversarial Networks

Boris Flach
Czech Technical University in Prague

- Generative adversarial networks
- Wasserstein distance for distributions
- Wasserstein GANs

Generative adversarial networks

As before with VAEs

- Let $\mathcal{X}$ denote the image space and $\mathcal{Z}$ denote a noise space. We may assume e.g. $\mathcal{Z} = \mathbb{R}^n$. We fix a simple distribution $Z \sim \mathcal{N}(0, \mathbb{I})$ on it.
- Given a sample $\mathcal{S}^m = \{x^j \in \mathcal{X} \mid j = 1, \ldots, m\}$, we want to learn a parametrised conditional distribution $p_w(x \mid z)$ represented by a mapping $g_w \colon \mathcal{Z} \to \mathcal{X}$ such that the joint model $p_w(x, z)$ is likely to generate images similar to those in $\mathcal{S}^m$.

In addition

- We consider a classifier network with a single output neuron. This network should discriminate natural images (coming from $\mathcal{S}^m$) from synthetic images generated by $p_w(x, z)$. Its output $d_v(x)$ is considered as probability of the class "natural".

Its loss is defined by

$$L(w, v) = \mathbb{E}_{x \sim \mathcal{S}^m}\Big[\log d_v(x)\Big] + \mathbb{E}_{z \sim p(z)}\Big[\log\big(1 - d_v \circ g_w(z)\big)\Big]$$

and the optimistation task is

$$\min_w \max_v L(v, w).$$

Simplified analysis of GANs

Let us analyse this saddle point task under the assumptions

- infinite training set $\mathcal{S}^m$, $m \to \infty$, denote the distribution by $p_d(x)$
- the generator model has infinite(!) capacity

The maximum w.r.t. the discriminator is achieved at $d^*(x) = \frac{p_d(x)}{p_d(x)+p_g(x)}$ because

$$L(g,d) = \int p_d(x) \log d(x)\, dx + \int p(z) \log\big(1 - d \circ g(z)\big)\, dz =$$

$$\int p_d(x) \log\big(d(x)\big) + p_g(x) \log\big(1 - d(x)\big)\, dx$$

Then the minimum w.r.t. the generator is achieved at $p_g = p_d$ because

$$C(g) = \max_d L(g,d) = \mathbb{E}_{x\sim p_d}\left[\log \frac{p_d(x)}{p_d(x)+p_g(x)}\right] + \mathbb{E}_{x\sim p_g}\left[\log \frac{p_g(x)}{p_d(x)+p_g(x)}\right]$$

Practical issues of GANs

Quite impressive results achieved by deep convolutional GANs, but there are technical problems

- what happens with $\log\big(1 - d(g(z))\big)$ early in learning, when g is poor?
- mode collapse: what happens if g maps all $z \in \mathcal{Z}$ on one x?
- searching saddle points of functions is complicated

Learning GANs is difficult and sensitive to hyper-parameter and architecture setup.

Conjecture: the main problem is comparing discrete distributions (training data) with continuous distributions (generator model).

Wasserstein distance for distributions

Many distances for distributions are problematic when

- one distribution is discrete and the other is absolutely continuous (has a density)
- the distributions have disjoint support

Consider the following situation: let $\mathbb{P}_r$ be a fixed distribution on $\mathcal{X}$, let Z be a random variable on $\mathcal{Z}$ and let $g_\theta \colon \mathcal{Z} \to \mathcal{X}$ be a parametrised family of mappings.

Denoting the distribution of $g_\theta(Z)$ by $\mathbb{P}_\theta$, we want a distribution distance $W(\mathbb{P}_r, \mathbb{P}_\theta)$ that behaves nicely as a function of θ.

Defintion 1. The Wasserstein distance of distributions $\mathbb{P}_1$ and $\mathbb{P}_2$ is defined by

$$W(\mathbb{P}_1, \mathbb{P}_2) = \inf_{\gamma \in \Pi(\mathbb{P}_1, \mathbb{P}_2)} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|,$$

where $\Pi(\mathbb{P}_1, \mathbb{P}_2)$ is the set of all joint distributions $\gamma(x, y)$ with marginals $\mathbb{P}_1$ and $\mathbb{P}_2$

Remarks:

- This distance is sometimes called earth-mover distance
- $\gamma(x, y)$ is interpreted as a transport plan and W is the cost of the optimal plan

Wasserstein distance for distributions

Theorem 1. *(w/o proof) Let g_θ be a neural network composed of affine transformations and smooth Lipschitz pointwise non-linearities. Let $p(z)$ be a prior over z with bounded expectation of its norm. Then $W(\mathbb{P}_r, \mathbb{P}_\theta)$ is continuous and almost everywhere differentiable in θ.*

How to compute the distance? The task is convex and its dual is

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}\big[f(x)\big] - \mathbb{E}_{x \sim \mathbb{P}_\theta}\big[f(x)\big],$$

where the supremum is over all 1-Lipschitz functions $f : \mathcal{X} \to \mathbb{R}$. Under some mild conditions on g_θ

- the problem $\max_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}\big[f(x)\big] - \mathbb{E}_{x \sim \mathbb{P}_\theta}\big[f(x)\big]$ has a solution and
- the gradient is $\nabla_\theta W(\mathbb{P}_r, \mathbb{P}_\theta) = -\mathbb{E}_{z \sim p(z)}\big[\nabla_\theta f(g_\theta(z))\big]$

Wasserstein GANs

- both, the generator mapping $g_\theta \colon \mathcal{Z} \to \mathcal{X}$ and the critic $f_w \colon \mathcal{X} \to \mathbb{R}$ are implemented as neural networks.
- f_w must be a 1-Lipschitz function (or at least a K-Lipschitz function).

In the inner loop of the algorithm (given the current θ) the critic is learned on x-batches sampled form the data and z-batches sampled from the prior noise distribution. An additional gradient norm penalty is used to enforce the Lipschitz requirement. The critic is learned till optimality.

In the outer loop θ is updated based on a z-batch sampled from $p(z)$ by using the gradient formula given above.