Generative Networks

Thomas Ricatte

2018-02-15

Outline

Motivation

Generative Networks

- A discriminative model is a way to model the conditional probability of a target Y (low-dimension) given some covariates X (high-dimension).
- Conversely, a generative model tries to model the conditional probability of X given Y (or even the joint probability $X \times Y$

Figure 1: Sampling from $P(X|Y)$ on MNIST using a ConditionalGan (Mirza and Osindero 2014)

Inverse convolutions

- Our objective is to expand the signal from a low-dimension representation to an high-dimension signal space.
- In feed-forward networks, the objective was to reduce the signal dimension using for instance conv layers

- To do the opposite, we introduce the inverse convolutional operator

Inverse convolutions (1D case)

1	4	-1	0	2

Conv

2	1

6	7	-2	2

Deconv

1	3

6	18			
	7	21		
		-2	-6	
			2	6

6	25	19	-4	6

$$\begin{pmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 2 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 & 0 & 0 & 0 \\ 0 & 1 & 3 & 0 & 0 \\ 0 & 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 & 3 \end{pmatrix}^T$$

Inverse convolutions

- Applying convolution + inverse convolution will keep the signal "roughly" unchanged
(intuition: mass of $K \cdot K^T$ will concentrate on the diagonal)
- We can define stride, padding and dilatation similarly to regular convolution
- Since it's an upscaling operation, it can creates artifacts on the resulting image especially when stride > 1

1	1	2	1	2	1	2	1	1

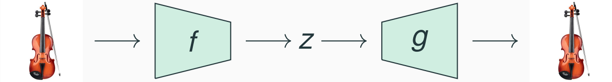
Figure 2: Result of $(1, 1, 1, 1) \circledast (1, 1, 1)$ (stride 2)

- In some cases, it's better to combine this with interpolation.

Variational Autoencoders

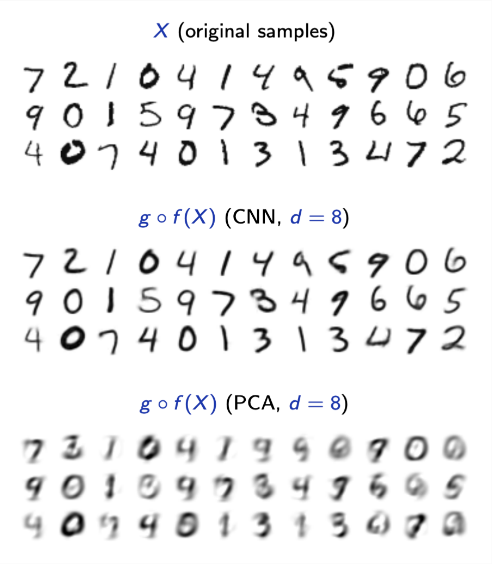
Autoencoders

- Main idea: force a self-supervised network to compress the original representation in a low-dimensional latent space.



- The goal is to learn an encoder f and a decoder g such that $g \circ f$ is close to identity.
- If f and g are linear, the optimal solution is given by a PCA
- Otherwise, we can achieve better performance with deep networks

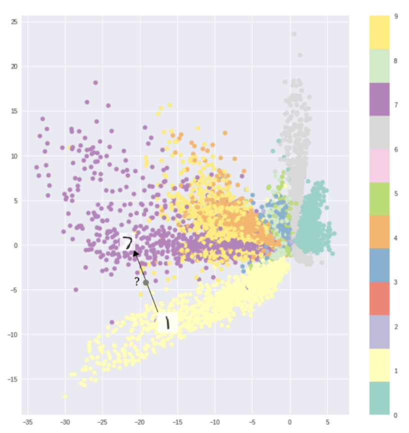
Deep Autoencoders



(by courtesy of François Fleuret)

How to sample from autoencoders ?

- Simple answer: sample z in the latent space and feed it into the decoder
- However it is very likely that the encoded inputs lies in a low-dimensional manifold inside the latent space



VAE in a nutshell

- **Let us constraint the latent variable z to follow a fixed distribution from which we can sample easily**
- Let's rewrite everything with probabilities !

$$x \longrightarrow p_\theta(z|x) \longrightarrow z \longrightarrow p_\theta(x|z) \longrightarrow x'$$

- $p_\theta(z|x)$ is untractable since we do not know the distribution of the true data so we approximate it by the variational distribution $q_\phi(z|x)$ that should minimize

$$\mathbb{D}_{KL}(q_\phi(z|x), p_\theta(z|x)) \quad .$$

Lemma

For any variational distribution q_ϕ, the (true) marginal log-likelihood $log(p_\theta(x))$ can be written as

$$\mathbb{D}_{KL}(q_\phi(z|x), p_\theta(z|x)) + \mathcal{L}_{\theta,\phi} \quad .$$

Note that:

- $\mathcal{L}_{\theta,\phi}$ is called the **variational lower bound** since $log(p_\theta(x)) > \mathcal{L}_{\theta,\phi}$
- For a fixed θ, minimizing the KL-divergence wrt ϕ is similar to **maximize** $\mathcal{L}_{\theta,\phi}$.
- For a fixed ϕ, maximizing $\mathcal{L}_{\theta,\phi}$ wrt θ, maximizes the marginal log-likelihood of the data.

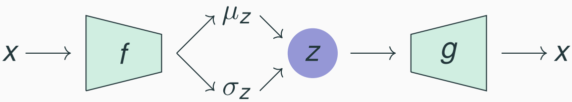
- Let's summarize ! The loss function to minimize is $-\mathcal{L}_{\theta,\phi}$ and can be rewritten as

$$\mathbb{E}_{z\sim q_\phi(z|x)}\left[-log(p_\theta(x|z))\right] + \mathbb{D}_{KL}(q_\phi(z|x)|p_\theta(z)) \quad .$$

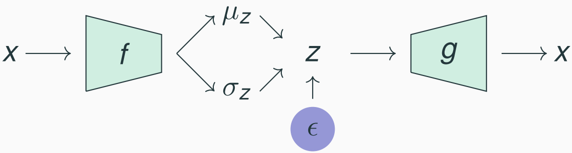
- The first term is called the reconstruction loss.
- The second term can be seen as a regularizer toward the prior distribution of the latent variable p_θ

One last problem ! How to backprop ?

- **Problem:** Impossible to backpropagate through a **stochastic node** like z



- **Solution:** Let's write $z = \mu_z + \sigma_z \odot \epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$ to have a differentiable path end-to-end.

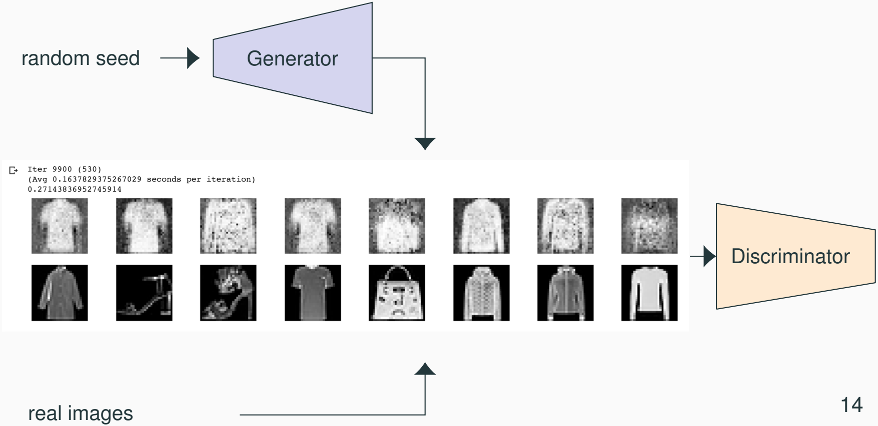


Reparametrization trick

Generative Adversarial Nets

GANs

New idea by Goodfellow et al. 2014: let us write the problem as a minimax game between a <u>generator</u> and a <u>discriminator</u>



GANs (cted)

- Let us consider a generator G parametrized by θ and a discriminator D parametrized by ϕ
  and
  - $(x^i)_{i=1\dots n}$ a batch of n training images
  - $(z^i)_{i=1\dots n}$ a batch of n noise samples sampled from a fixed noise prior.
- The goal of the discriminator is to distinguish between $G(z)$ and x so minimize the negative log-likelihood

$$NLLH(x, z, \theta) = -\left[\sum_{i=1}^{n} log(D_\theta(x^i)) + log(1 - D_\theta(G_\phi(z^i))\right] .$$

- The goal of the generator is to minimize the log-likelihood

$$LLH(x, \phi) = \sum_{i=1}^{n} log(1 - D_\theta(G_\phi(z^i))$$

GANs :: Pathological behaviors

- **Oscillation / bad convergence**
  Due to minimax game
- **Mode collapse**
  Happens when the training data is multi-modal (which is usually the case in practice): can be a good strategy for the generator to target the easiest mode of the target distribution (pullover in the example below)

GANs :: Alchemy ?

Lots of "hacks" to stabilize the training

1. Normalize the inputs
2. $min\,log(1 - D)$ vs $max\,log(D)$
3. Choose the noise prior wisely
4. BatchNorm on full real / fake images
5. Avoid Sparse Gradients (ReLu -> LeakyReLu)
6. Use soft / noisy labels
7. Choose the optimizers wisely (e.g. Adam for G, SGD for D)
8. ...

(from https://github.com/soumith/ganhacks)

GANs :: (A bit of) theory

- Let us denote
  - μ the density of the true data
  - $\mu_G = G(\mu_{\text{noise}})$ the density of the data generated by a generator G
- Our main goal is to find G that minimizes the distance between μ and μ_G
- **Intuition**: the bigger gap between μ and μ_G, the better the optimal discriminator.

Can we formalize this intuition ?

Theorem

The optimal discriminator (without regularization) D_G^* is

$$x \to \frac{\mu(x)}{\mu(x) + \mu_G(x)} \quad .$$

The corresponding loss at this point is

$$\mathcal{L}_G(D_G^*) = 2\mathbb{D}_{JS}(\mu, \mu_G) - log(4) \quad ,$$

where $\mathbb{D}_{JS}$ is the Jensen-Shannon divergence (symmetric variant of the KL-divergence).

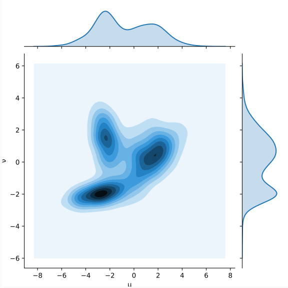
Training the GAN $\equiv$ finding G that minimizes $\mathbb{D}_{JS}(\mu, \mu_G)$

Wasserstein GANs

- Arjovsky et al. 2017 claims that the Jensen-Shannon divergence does not allow to take into account the metric structure of the space.
- They proposes to go with the Wasserstein distance $\mathbb{D}_{W_1}$.

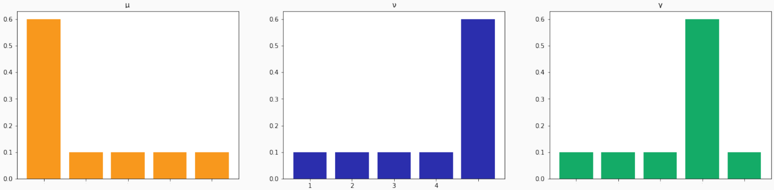
$$\mathbb{D}_{W_1}(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \int d(x, y) d\gamma(x, y)$$

- "earth moving distance"



Wasserstein GANs (cted)

Advantages of $\mathbb{D}_{W_1}$ over $\mathbb{D}_{JS}$?



$$\mathbb{D}_{W_1}(\mu, \nu) = 2 > \mathbb{D}_{W_1}(\mu, \gamma) = 1.5$$

$$\mathbb{D}_{JS}(\mu, \nu) = 0.20 < \mathbb{D}_{JS}(\mu, \gamma) = 0.25$$

Problem: How to compute $argmin_G \, \mathbb{D}_{W_1}(\mu, \mu_G)$?

Wasserstein GANs (cted)

- Using Kantorovich-Rubinstein duality theorem,

$$\mathbb{D}_{W_1}(\mu, \mu_G) = \max_{\|D|_L \leqslant 1} \left[\mathbb{E}_{X \sim \mu}\left[D(X)\right] - \mathbb{E}_{X \sim \mu_G}\left[D(X)\right]\right] \quad ,$$

where $\|D|_L$ is the Lipschitz semi-norm equal to

$$\max_{x,y} \frac{\|D(x) - D(y)\|}{\|x - y|} \quad .$$

- We get a **new loss** for the discriminator !
- Main issue is to deal with the semi-norm constraint !
    - Weight clipping (original idea)
    - Smooth penalty (Gulrajani et al. 2017)

Conditional GANs

Cycle GANs

Image 2 Image

Wrapping up

VAE vs GANs

	VAE	GAN
Modules	Encoder + Decoder	Generator + Discriminator
Training ?	Reconstruction Loss + Latent Loss	Minimax game
Stability ?	Closed-form	Need to reach a Nash equilibrium
Quality ?	Good but blurry images	High quality sharp images

References

References

- Arjovsky, Martin et al. (2017). “Wasserstein gan”. In: arXiv preprint arXiv:1701.07875.
- Goodfellow, Ian et al. (2014). “Generative adversarial nets”. In: Advances in neural information processing systems, pp. 2672–2680.
- Gulrajani, Ishaan et al. (2017). “Improved training of wasserstein gans”. In: Advances in neural information processing systems, pp. 5767–5777.
- Mirza, Mehdi and Simon Osindero (2014). “Conditional Generative Adversarial Nets”. In: CoRR abs/1411.1784. arXiv: 1411.1784. URL: http://arxiv.org/abs/1411.1784.