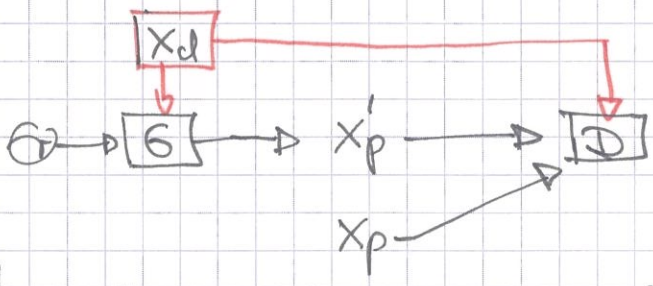
Solution?

Use a conditional structure instead! [cGAN [1411.1784]]



~~keep~~

↳ we ~~keep~~ restore locality be feeding $(x_d, x_p^{(\prime)})$ tuples to the discriminator → sees if x'_p is too far away from x_d!

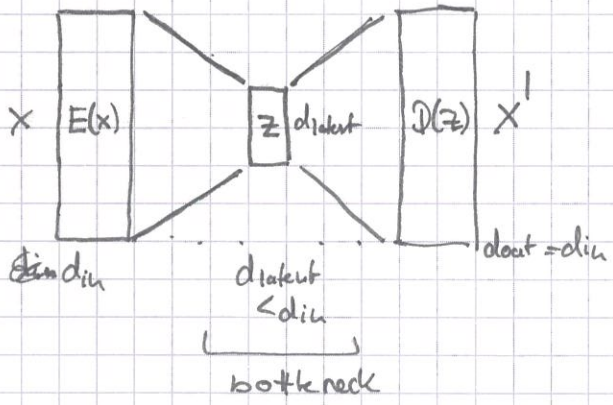
↳ additional random noise into G → make mapping stochastic!

↳ show plot! → now cuts work!

→ still not great → we will see later NF
↳ no explicit unfolding distribution! → need proper dkdat?!

2.3. Variational Autoencoder (VAE) [1312.6114]

AE:



loss: MSE

$$L = \frac{1}{N} \sum |x_i - x'_i|$$

$$x = D(E(x))$$

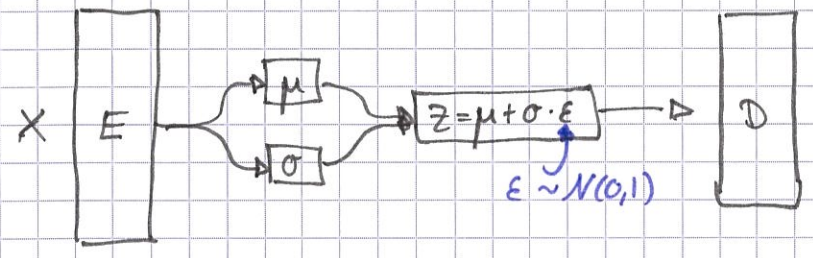
- compress info (learn most useful latent rep of data)

→ generative model: sample z + use D(z)

but: from which z should we sample?

latent-space is not regularized!

VAE gives structure to latent space



want $E(z|x) \overset{!}{=} p(z|x)$! match

→ Now we can sample from VAE! → $E(z) \to E(z|x) \equiv q(z|x)$

skip derivation of Loss here, ~~but~~ $D(x) \to D(x|z) \equiv p(x|z)$

- ↳ Loss [2211.01421]+[1312.6114]

$$L_{VAE} = \underbrace{\langle -\log D(x|z) \rangle_{z \sim E(z|x)}}_{\text{becomes MSE loss if decoder output/prob is a gaussian with constant width}} + \underbrace{D_{KL}[E(z|x), p_z(z)]}_{\text{forces latent-distribution } E(z|x) = q(z|x) \text{ to match prior}}$$

$L_{VAE} \Leftrightarrow K_{KL}(q(z|x), p(z|x))$

→ similar to L_{BNN}! now sampling over z not θ!

pros:
- \+ easy to train
- \+ explicit map to latent-space
- \+ explicit distribution $q(z|x)$

- − rarely state of the art
- − no optimality guaranteed
- − likelihood gap! missing $p(x)$

→ can be linked to flows [2007.02731]

↳ example: [2101.08944]

↳ inversion ~ okish!

$\log p(x)$ ~ not tractable!

↳ next slide

III. Normalizing Flows

↳ GANs: no info about $p(x)$ → only implicitely!

VAE: only ELBO of $p(x)$:

as:

$$\log p(x) = \int dz\, q(z|x) \log p(x) \quad \text{(add 1!)}$$

$$= \int dz\, q(z|x) \log \frac{p(x|z)p(z)}{p(z|x)} \quad \text{(Bayes)}$$

$$= \int dz\, q(z|x) \left[\log p(x|z) - \log \frac{q(z|x)}{p(z)} + \log \frac{q(z|x)}{p(z|x)} \right]$$

$$= \langle \log p(x|z) \rangle_{q(z|x)} - KL[q(z|x), p(z)] + KL[q(z|x), p(z|x)]$$

$$\geq \langle \log (p(x|z) \rangle_{q(z|x)} - KL[q(z|x), p(z)]$$

(last KL: match from VAE)

→ in contrast: Flow!

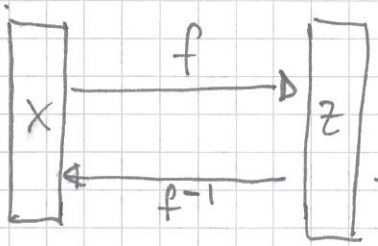
Can be rewritten as:

$$(*) \quad = \left\langle \log p(z) + \underbrace{\log \frac{p(x|z)}{q(z|x)}}_{V(x,z)} + \underbrace{\log \frac{q(z|x)}{p(z|x)}}_{\mathcal{E}(x,z)} \right\rangle_{z \sim q(z|x)}$$

for VAE: $\mathcal{E}(x,z) > 0$

Now: for normalizing flows we can learn p expicitely!

key idea:



learn invertible map btw data & simple nois → $\dim z \stackrel{!}{=} \dim x$

→ $p(x)\,dx = p(z)\,dz \qquad z = f(x) \leftrightarrow x = f^{-1}(z)$

↳ $$p(x) = p(z = f(x) \cdot \left| \frac{\partial f}{\partial x} \right|$$

⇒ f = Density estimation

f^{-1} = generation!

Now: What is the link to (*)?

$$\Rightarrow \log p(x) = \mathbb{E}\left[\log p(z) + \log\left|\frac{\partial f}{\partial x}\right|\right]$$

$$= \int dz\, q(z|x)\left[\log p_z(z) + \log\frac{p(x|z)}{q(z|x)} + \underbrace{\log\frac{p(z|x)}{q(z|x)}}_{=0}\right]$$

now invertible means: $q(z|x) = p(z|x) = \delta(z - f(x))$

$$p(x|z) = \delta(x - f^{-1}(z))$$

$$= \log p_z(f(x)) + \log\left|\frac{\partial z}{\partial x}\right|$$

⇒ proof or check [2007.02731]

Now: we need a tractable jacobian $\left|\frac{\partial f}{\partial x}\right|$

↳ Loss NF: $\boxed{\mathcal{L}_{NF} = -\log p_\theta(x)}$

↳ linked to KL divergence $KL(p(x), p_\theta(x)) = \int dx\, p(x) \cdot \log\frac{p(x)}{p_\theta(x)}$

$$= \int dx\, p(x)\log(x) - \int dx \log p_\theta(x)$$

$$\simeq -\langle \log p_\theta(x)\rangle_{x\sim p(x)} + \text{const}$$

→ more stable, more powerful!

3.1. Tractable jacobian

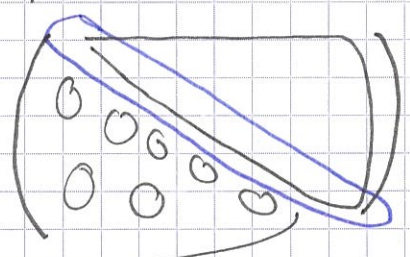
$\left|\frac{\partial f}{\partial x}\right|$ ← det of $d \times d$ matrix

↳ scales like $O(d^3)$ ← bad

How to fix this?

Basic idea: Autoregressive transformations

$$z_1 = z_1(x_1)$$
$$z_2 = z_2(x_1, x_2)$$
$$\vdots$$
$$z_n = z_n(x_1, \ldots, x_n)$$



↳ det ~ $O(d)$

→ tractable ✓

In the end, flow is a chain of multiple such block:

$$x \to z^{(1)} \to z^{(2)} \to \dots \to z^{(n)} \} \text{ NF}$$

↑ permutations to change meaning of z_i

↑ is a rotation → bijection with $|\det J| = 1$

~~Downsides of Autoregressive flows: fast in direction~~

How to construct these function $z_i = f_i(x_1, \dots, x_i)$ such that they are invertible?

1) Affine transf. couplings: → RealNVP [1605.08803] INN [1808.04730]

$$z_i = \alpha_i(x_1, \dots, x_{i-1}) \cdot x_i + \mu_i(x_1, \dots, x_{i-1}) \quad \to \text{NN!}$$

$$\Rightarrow \boxed{\begin{aligned} z_1 &= \alpha_1 \cdot x_1 + \mu_1 \\ z_2 &= \alpha_1(x_1) \cdot x_2 + \mu_2(x_1) \end{aligned}} \to \text{check invertible!}$$

$$\text{inverse: } \boxed{x_i = \frac{z_i - \mu_i}{\alpha_i}} \to \text{in practice } \alpha = e^{NN(x)}$$

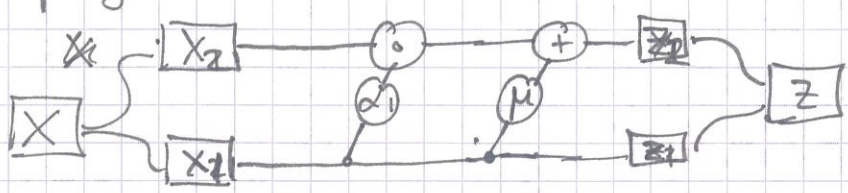
but

$$\left.\begin{aligned} x_1 &= \frac{z_1 - \mu_1}{\alpha_1} \\ x_2 &= \frac{z_2 - \mu_2(x_1)}{\alpha_1(x_1)} \end{aligned}\right\} \Rightarrow \text{inverse direction is } d \text{ times slower!}$$

other options:

3.2. Coupling blocks

Instead of autoregressive transformation, we can also consider coupling blocks



$$\boxed{\begin{aligned} z_1 &\equiv x_1 \\ z_2 &= \alpha(x_1) \cdot x_2 + \mu(x_1) \end{aligned}}$$

Inverse:

$$\boxed{\begin{aligned} x_1 &= z_1 \\ x_2 &= \frac{z_2 - \mu(z_1)}{\alpha(z_1)} \end{aligned}}$$

→ FAST

Beyond AFFINE-Blocks:

- piecewise splines:
  - linear, quadratic [1808.03856]
  - cubic splines [1906.02145]
  - rational quadratic [1906.04032]
    - ↳ $C = \alpha$

3.3. Example I: Event generation [2110.13632]

They looked at slighly different dataset:

$$pp \to Z(\to \mu\mu) + \{1,2,3\} \text{ jets}$$

~~→ cubic spli~~

→ INN variant → fast training & sampling

↳ show plots

→ much better results → more stable training

3.4. Example II: Unfolding [2006.06685]

Ansatz is that unfolding can be written as:

$$\frac{d\sigma}{dx_d} = \int dx_p \; \frac{d\sigma}{dx_p} \; f(x_d, x_p) \quad \leftarrow \text{transfer function}$$

with inverse:

$$\frac{d\sigma}{dx_p} = \int dx_d \; \frac{d\sigma}{dx_d} \; \bar{f}(x_p, x_d) \quad \leftarrow \text{unfolding}$$

$$\Rightarrow \int dx_d \; \bar{f}(x_p, x_d) f(x_d, x_p') = \delta(x_p - x_p')$$

Now, we want things to be stochastic in nature, so

$$\to \; \bar{f}(x_d|x_p) \in p(x_p|x_d)$$

$$\to \; p(x_p|x_d) = \frac{p(x_d|x_p) \cdot p(x_p)}{p(x_d)}$$

learn this:

training: $\min -\log p(x_p|x_d)$

$$z \to \text{cINN}^{x_d} \to x_d$$

→ ⇒ we can sample it

↳ plots

+ good calibrat
+ single unfoldin
↓
+ better than GA