

Generative models

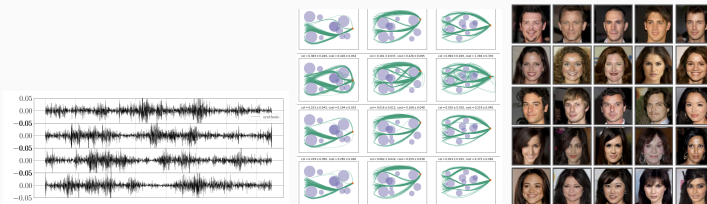
how they work and how to train them

Simon Coste

September 20, 2023

Intro: Generative Modelling

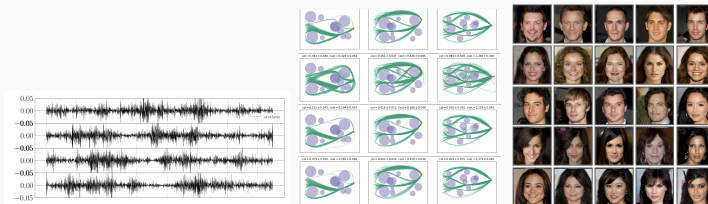
x_*^1, \dots, x_*^n : training samples from an unknown distribution ρ_* (“target”)



The two goals of generative modelling:

1. Generate ‘new’ samples from ρ_* (direct problem)
2. Find a ‘good’ estimator $\hat{\rho}_*$ for ρ_* (inverse problem)

x_*^1, \dots, x_*^n : training samples from an unknown distribution ρ_* (“target”)



The two goals of generative modelling:

1. Generate ‘new’ samples from ρ_* (direct problem)
2. Find a ‘good’ estimator $\hat{\rho}_*$ for ρ_* (inverse problem)

EBMs, GANs, VAEs, Normalizing Flows, Neural ODEs, Noise Contrastive Estimation, Diffusions, Flow matching, Consistency models

Energy-Based Models

$U_\theta : \mathbb{R}^d \rightarrow \mathbb{R}_+ =$ parametrized family of functions (“model energies”)

Definition of the model densities:

$$\rho_\theta(x) = \frac{e^{-U_\theta(x)}}{Z_\theta} \quad Z_\theta = \int e^{-U_\theta(x)} dx.$$

EBM training = finding θ_* achieving the best ‘fit’ between ρ_θ and ρ_*

$$\theta_* \in \arg \min \text{dist}(\rho_*, \rho_\theta)$$

Using an EBM

Once U_{θ_*} has been trained, new synthetic samples are obtained by sampling from the distribution

$$\hat{\rho}_* = \rho_{\theta_*} = \frac{e^{-U_{\theta_*}}}{Z_{\theta_*}}.$$

This step typically needs MCMC methods such as Langevin:

$$X_{\tau+1} = X_{\tau} - \eta \nabla_x U_{\theta_*}(X_{\tau}) + \sqrt{\eta} \xi_{\tau} \quad \xi_{\tau} \sim \mathcal{N}(0, I)$$

This is called “implicit generation” [Du and Mordatch 19].

Advantages of EBM

- **Simplicity.** Only one neural network U_θ
→ *VAEs and GANs require at least two!*
- **Flexibility.** We can exploit the tradeoff between quality and cost
→ *impossible with feed-forward generators such as GANs or NFs*
- **Compositionality.** Combining different EBMs is as simple
→ *just add the energies*
- **Reusability.** Can be used to help various other tasks
→ *inpainting, importance sampling, OOD detection...*

Choosing the right loss for EBM learning

Kullback-Leibler divergence \leftrightarrow max-likelihood

$$\begin{aligned}\text{dist}(\rho_\theta, \rho_*) &= \mathbb{E}_{X \sim \rho_*} \log \rho_*(X) - \log \rho_\theta(X) \\ &\approx \text{cst} - \frac{1}{n} \sum \log \rho_\theta(x_i^*)\end{aligned}$$

Fisher divergence

$$\begin{aligned}\text{dist}(\rho_\theta, \rho_*) &= \mathbb{E}_{X \sim \rho_*} |\nabla \log \rho_*(X) - \nabla \log \rho_\theta(X)|^2 \\ &\approx \frac{1}{n} \sum |\nabla \log \rho_*(x_i^*) - \nabla \log \rho_\theta(x_i^*)|^2\end{aligned}$$

Training procedures

I: max-likelihood

Gradient ascent on Energy-Based Models

Goal: maximize $L(\theta) = \mathbb{E}_*[\log \rho_\theta] = -\mathbb{E}_*[U_\theta + \log Z_\theta]$

$$\nabla L(\theta) = -\mathbb{E}_*[U_\theta] - \nabla \log Z_\theta$$

Computation of $\nabla_\theta \log Z_\theta$:

$$\frac{\nabla_\theta Z_\theta}{Z_\theta} = \int -\nabla_\theta U_\theta(x) e^{-U_\theta(x)} \frac{1}{Z_\theta} dx = -\mathbb{E}_\theta[\nabla_\theta U_\theta]$$

Gradient ascent with stepsize $\eta > 0$

$$\theta_{t+1} - \theta_t = \eta \times (\mathbb{E}_{\theta_t}[\nabla_\theta U_{\theta_t}] - \mathbb{E}_*[\nabla_\theta U_{\theta_t}])$$

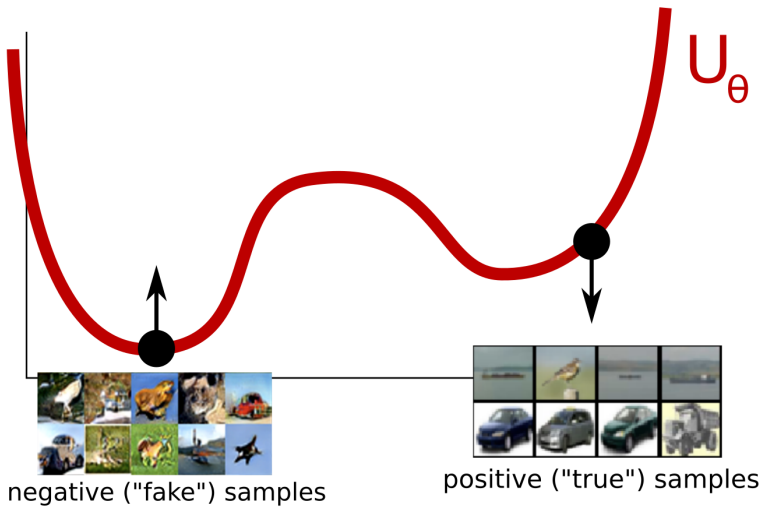
$$\nabla_{\theta} L(\theta) = \nabla_{\theta} (\mathbb{E}_{*}[U_{\theta}(X)] - \mathbb{E}_{\theta}[U_{\theta}(X)])$$

$$\begin{aligned} & \mathbb{E}_{*}[U_{\theta}] \\ x_i^{*} = & \text{"positive samples"} \\ & \text{from } \rho_{*} \\ & \approx \frac{1}{n} \sum_i U_{\theta_t}(x_i^{*}) \end{aligned}$$

$$\begin{aligned} & \mathbb{E}_{\theta_t}[U_{\theta}] \\ y_i = & \text{"negative samples"} \\ & \text{from } \rho_{\theta_t} \\ & \approx \frac{1}{n} \sum_i U_{\theta_t}(y_i) \end{aligned}$$

"contrastive learning" :

- pull down the energy of positive samples, $\mathbb{E}_{*}[U_{\theta}]$
- pull up the energy of negative samples, $\mathbb{E}_{\theta_t}[U_{\theta}]$



MCMC sampling is too costly

Q: at each gradient step, how do we get the negative samples for computing $\mathbb{E}_{\theta}[U_{\theta}]$?

A: using MCMC/Langevin methods...

At step t , initialize X_0^i (“walkers”), then for $\tau = 0, \dots, T_{mix}$,

$$X_{\tau+1}^i = X_{\tau}^i - \eta \nabla U_{\theta}(X_{\tau}^i) + \sqrt{2\eta} \xi_{\tau}$$

and estimate

$$\mathbb{E}_{\theta_t}[U_{\theta(t)}] \approx \frac{1}{N_{walkers}} \sum_{i=1}^{N_{walkers}} U_{\theta_t}(X_{T_{mix}}^i).$$

If T_{mix} is large, this is too costly.

Each gradient ascent step will consume T_{mix} MCMC sampling steps for each of the $N_{walkers}$ chains!

Contrastive Divergence with k steps (CD- k), [Hinton 2005]

- don't let the chain reach T_{mix} steps. Use only k steps ($k = 1$).
- initialize each chain directly at the training points $\{x_*^i\}$.

Contrastive Divergence with k steps (CD- k), [Hinton 2005]

- don't let the chain reach T_{mix} steps. Use only k steps ($k = 1$).
- initialize each chain directly at the training points $\{x_*^i\}$.

[Hyvarinen 2007]

in the limit of small noise $\eta \rightarrow 0$, CD-1 = score matching.

[Yair and Michaeli 20] CD-1 is an adversarial game

[Agoritsas et al 23] Effect of non-convergent sampling

Persistent Contrastive Divergence (PCD), [Tieleman 2008]

- don't let the chain reach T_{mix} steps. Use only k steps ($k = 1$).
- ~~Initialize each chain directly at the training points $\{x_*^i\}$.~~
- initialize each chain directly where the previous chain ended.

Persistent Contrastive Divergence (PCD), [Tieleman 2008]

- don't let the chain reach T_{mix} steps. Use only k steps ($k = 1$).
- ~~Initialize each chain directly at the training points $\{x_*^i\}$.~~
- initialize each chain directly where the previous chain ended.

Practically: maintain a set of *walkers* X_t^i . At step $t + 1$,

- 1) approximate $\mathbb{E}_{\theta_t}[U_{\theta_t}] \approx \frac{1}{n} \sum_{i=1}^N U_{\theta_t}(X_t^i)$,
- 2) compute θ_{t+1} using the approximation,
- 3) move the walkers with $X_{t+1} = X_t - \eta \nabla U_{\theta_{t+1}}(X_t) + \sqrt{2\eta} \xi$

\Rightarrow leads to mode collapse.

Replay buffer techniques [Du and Mordatch 2019]

- don't let the chain reach T_{mix} steps. Use only k steps ($k = 1$).
- ~~Initialize each chain directly at the training points $\{x_*^i\}$.~~
- ~~initialize each chain directly where the previous chain ended.~~
- initialize, sometimes from the past, sometimes from pure noise

+ many other tricks.

Training procedures

II: alternative losses

- a Noise Contrastive methods
- b GANs
- c Score Matching
- d Denoising score matching

a. Noise Contrastive Estimation [Gutmann & Hyvarinen 2010]

Let μ be a known density and $y_i \sim \mu$ be iid.

Idea: train a binary classifier with logistic regression to distinguish between **true** samples x_i^* and **fake** samples y_i .

Bayes' rule gives the optimal classifier D_{opt} :

$$\begin{aligned} D_{\text{opt}}(x) = \mathbb{P}(\text{true} \mid x) &= \frac{p(x \mid \text{true})}{p(x \mid \text{fake}) + p(x \mid \text{true})} \\ &= \frac{\rho_*(x)}{\rho_*(x) + \mu(x)} \end{aligned}$$

a. Noise Contrastive Estimation [Gutmann & Hyvarinen 2010]

Let μ be a known density and $y_i \sim \mu$ be iid.

Idea: train a binary classifier with logistic regression to distinguish between **true** samples x_i^* and **fake** samples y_i .

Bayes' rule gives the optimal classifier D_{opt} :

$$\begin{aligned} D_{\text{opt}}(x) &= \mathbb{P}(\text{true} \mid x) = \frac{p(x \mid \text{true})}{p(x \mid \text{fake}) + p(x \mid \text{true})} \\ &= \frac{\rho_*(x)}{\rho_*(x) + \mu(x)} \end{aligned}$$

Goal: maximize $R(\theta) = \sum_{i=1}^n \log D_{\theta}(x_i^*) + \log(1 - D_{\theta}(y_i))$.

Set $D_{\theta}(x) = F_{\theta}(x)/(F_{\theta}(x) + \mu(x))$ with $F_{\theta}(x) = e^{-U_{\theta}(x)+c_{\theta}}$.

If $D_{\theta_*} \approx D_{\text{opt}}$ then $F_{\theta_*} \approx \rho_*$

The normalization $\int e^{-U_{\theta}+c_{\theta}} = 1$ is automatic!

b. Limitations of NCE and GANs

- if μ is too close to ρ_* then training the classifier is too difficult
- if μ is too different to ρ_* then classifying is too easy, there are near-optimal classifiers very different than the optimal one

the GAN idea

\Rightarrow also train a "fake sample generator", say μ_β , instead of using always the same fixed generator μ

GAN objective: $\min_{\theta} \max_{\beta} \mathbb{E}_{*}[\log D_{\theta}] + \mathbb{E}_{\mu_{\beta}}[\log(1 - D_{\theta})]$

Min-Max problems are hard.

c. Score Matching [Hyvarinen 2005]

Goal: minimize $SM(\theta) = \mathbb{E}_* [|\nabla \log \rho_\theta - \nabla \log \rho_*|^2]$.

Hyvarinen 2005

$$SM(\theta) = \text{cst} + \mathbb{E}_* [|\nabla \log \rho_\theta|^2 + 2\Delta \log \rho_\theta].$$

- Parametrize the score $\nabla \log \rho_\theta$ with a neural network s_θ
- Minimize $\mathbb{E}_* [|s_\theta|^2 + 2\nabla_x \cdot (s_\theta)]$ using gradient descent

Problem 1: for $\nabla_\theta SM(\theta)$ we need to compute "double derivatives" like

$$\nabla_\theta \nabla_x \cdot s_\theta(x).$$

Problem 2: inferring $\log \rho$ from $s_\theta \approx \nabla \log \rho_\theta$?

Proof of Hyvarinen's identity: it's just an integration by parts.

For p, q two smooth densities with fast decay at ∞ ,

$$\begin{aligned}\mathbb{E}_p[|\nabla \log p - \nabla \log q|^2] &= \int p |\nabla \log p - \nabla \log q|^2 \\ &= \int p |\nabla p/p - \nabla q/q|^2 \\ &= c_p + \int p |\nabla / q|^2 - 2 \int \nabla p \cdot \nabla \log q \\ &= c_p + \int p |\nabla \log q|^2 + 2 \int p \nabla \cdot \nabla \log q \\ &= c_p + \mathbb{E}_p[|\nabla \log q|^2 + 2\Delta \log q]\end{aligned}$$

d. Denoising Score Matching [Vincent 2009]

Let us corrupt the original samples with noise:

$$y_i^* = x_i^* + \epsilon_i \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

distribution of $y_i^* = \rho_* * \mathcal{N}$.

Vincent 2009

$$SM(\theta) = \text{cst} + \mathbb{E}_{X \sim \rho_*, \epsilon \sim g}[|\nabla \log g(\epsilon) - \nabla \log \rho_\theta(X + \epsilon)|^2].$$

- Parametrize the score of the noisy distribution with s_θ (NN)
- Minimize $\mathbb{E}[|\epsilon/\sigma - s_\theta(X + \epsilon)|^2]$
- \Rightarrow no double derivatives!

- 1) In the presence of high energy barriers, SM methods and variants cannot learn the relative weights of the modes and/or lead to mode collapse.
- 2) The score $s_{\theta_*} \approx \nabla \log \rho_*$ does not give direct access to the density!
- 3) Results were good... but not as good as GANs

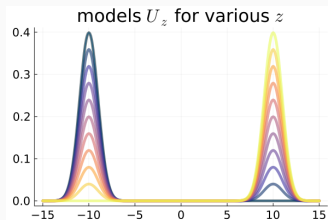
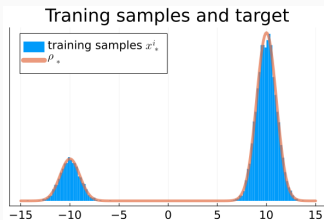
Training procedures III: some insights from toy models

Model: all gaussian mixtures with modes $a = -10, b = 10$:

$$U_z(x) = -\log \left(e^{-|x-a|^2/2} + e^{-z} e^{-|x-b|^2/2} \right)$$

$$Z_z = (1 + e^{-z})\sqrt{2\pi}$$

$$\rho_z(x) = \frac{e^{-|x-a|^2/2} + e^{-z} e^{-|x-b|^2/2}}{(1 + e^{-z})\sqrt{2\pi}}$$



Target: $\rho_* = \rho_{z_*}$ for some z_* with $q_* = \frac{e^{-z_*}}{1+e^{-z_*}} \approx 0.8$.

Useful approximations

$$\begin{aligned}\nabla_x U_z(x) &= \frac{(x-a)e^{-(x-a)^2/2} + e^{-z}(x-b)e^{-(x-b)^2/2}}{e^{-(x-a)^2/2} + e^{-z}e^{-(x-b)^2/2}} \\ &\approx (x-a)1_{x \text{ close to } a} + (x-b)1_{x \text{ close to } b}\end{aligned}$$

$$\nabla_z U_z(x) = e^{-z}e^{-(x-b)^2/2}/U_z(x) \approx 1_{x \text{ is close to } b}$$

$$\forall z, w \quad \mathbb{E}_w[\nabla_z U_z] \approx \mathbb{P}_w(\text{mode } b) = \frac{e^{-w}}{1+e^{-w}}$$

Gradient flow (continuous version of the discrete gradient descent):

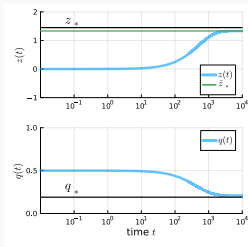
$$\dot{z}(t) = -\nabla_z \text{loss}(\rho_*, \rho_{z(t)})$$

where `loss` is one of the various objectives above.

Success of max-likelihood

$$\begin{aligned}\dot{z}(t) &= \mathbb{E}_{z(t)}[\nabla_z U_{z(t)}] - \mathbb{E}_{z_*}[\nabla_z U_{z(t)}] \\ &\approx \frac{e^{-z(t)}}{1 + e^{-z(t)}} - \frac{e^{-z_*}}{1 + e^{-z_*}}.\end{aligned}$$

Clearly this system converges towards its unique FP $z(t) = z_*$.



Failure of score matching

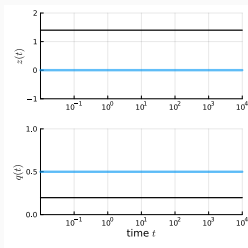
$$\dot{z}(t) = -\nabla_z SM(z) = \nabla_z \mathbb{E}_{z_*} [|\nabla \log \rho_{z(t)} - \nabla \log \rho_{z_*}|^2]$$

Remember that

$$\nabla \log \rho_z(x) \approx (x - a)1_{x \text{ close to } a} + (x - b)1_{x \text{ close to } b}$$

$\Rightarrow \nabla \log \rho_z(x)$ does not depend on z , hence $\nabla_z SM(z) \approx 0$.

This leads to the “no learning” phenomenon $\dot{z}(t) \approx 0$



Mode collapse in PCD

Here the negative samples are generated using

$$dX_t = -\nabla_x U_{z(t)}(X_t)dt + \sqrt{2}dB_t$$

Remember that

$$\nabla_x U_z(x) \approx (x - a)1_{x \text{ close to } a} + (x - b)1_{x \text{ close to } b}$$

X_t close to $b \Rightarrow dX_t \approx -(X_t - b)dt + \sqrt{2}dB_t$: Ornstein-Uhlenbeck

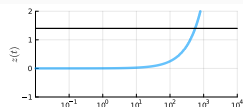
X_t close to $a \Rightarrow dX_t \approx -(X_t - a)dt + \sqrt{2}dB_t$: Ornstein-Uhlenbeck

There is no transfer of walkers between modes a and b !

The distribution of X_t does not change and is equal to $\rho_{z(0)}$:

$$\dot{z}(t) \approx \frac{e^{-z(0)}}{1 + e^{-z(0)}} - \frac{e^{-z_*}}{1 + e^{-z_*}} = \text{cst}$$

This leads to mode collapse, $z(t) \rightarrow \pm\infty$.



Thanks for the invitation!



BNP PARIBAS



École des Ponts
ParisTech



FONDATION
Université Paris Cité

Some references (with links)

How to train your EBMs (Song & Kingma)

Score Matching (Hyvarinen)

Denoising score matching (Vincent)

Noise contrastive estimation (Gutman and Hyvarinen)

Conditional NCE (Ma and Collins)

Improved CD (Du et al.)

Implicit generation (Du et al.)

Reduce, Reuse, Recycle (Du et al.)

Efficient training of EBMs (Carbone et al.)

From SM to diffusion models (Song and Ermon)