Adversarial Variational Bayes

Mauro Camara Escudero

School of Mathematics, University of Bristol

Background

The Inference and Generative Problems

$\mathbf{x}$ data, $\mathbf{z}$ latent variables, $\boldsymbol{\theta}$ parameters.

- Aim 1: Posterior inference with $p(\mathbf{z} \mid \mathbf{x})$
- Aim 2: Generative modelling i.e. $\mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x})$

Problem: the marginal likelihood is intractable due to marginalization

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z}$$

making both inference and generative modelling difficult!

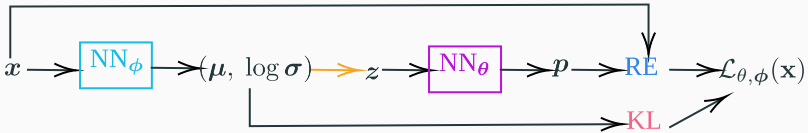
- Variational Auto-Encoders (VAEs) can perform approximate posterior inference and we get a generative model as a by-product
- Generative Adversarial Networks (GANs) only perform generative modelling, but often produce much sharper results than VAEs.

VAEs

VAEs optimize the following variational lower bound to the log likelihood.

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) \geq \mathcal{L}_{\theta,\phi}(\mathbf{x}) = \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})\right]}_{\text{Reconstruction Error (RE)}} - \underbrace{\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \mid\mid p(\mathbf{z}))}_{\text{Regularization Term (RT)}}$$

Define an encoder network to parametrize $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$, approximating the posterior $p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})$. Also define a decoder network parametrizing $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$.



Encoder and Decoder's parameters $(\boldsymbol{\phi}, \boldsymbol{\theta})$ are trained jointly using SGD. Back-propagation can compute $\nabla_{\boldsymbol{\phi}}$ thanks to the reparametrization trick

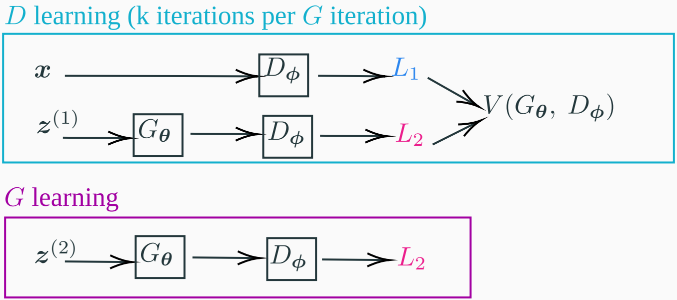
$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon} \qquad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}).$$

GANs

GANs play a minimax game (transformed to double-loop maximization)

$$\max_{\boldsymbol{\theta},\boldsymbol{\phi}} V(G_{\boldsymbol{\theta}}, D_{\boldsymbol{\phi}}) = \max_{\boldsymbol{\theta},\boldsymbol{\phi}} \underbrace{\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\log D_{\boldsymbol{\phi}}(\mathbf{x})\right]}_{L_1} + \underbrace{\mathbb{E}_{p(\mathbf{z})}\left[\log D_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\mathbf{z}))\right]}_{L_2}$$

between a generator network $G_{\boldsymbol{\theta}}$, trying to generate realistic data, and a discriminator $D_{\boldsymbol{\phi}}$, trying to accurately classify training and generated data.



VAEs vs GANs

- VAEs use an inference network to aid generative learning. This means we have access to $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ at the end.
- GANs use a discriminator network to aid generative learning. This means that we can't perform posterior inference.
- GANs tend to generate ”sharper”, more realistic results.
- In GANs we need to make sure that in the inner loop $D_{\boldsymbol{\phi}}$ is near-optimal before optimizing $G_{\boldsymbol{\theta}}$ in the outer loop.

Combining VAEs and GANs

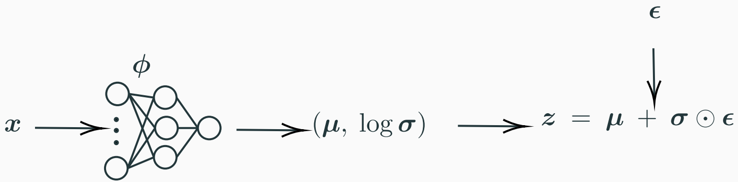
Implicit Approximate Posterior (I)

In a vanilla VAE $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x})\mathbf{I})$ has an analytical form. For this reason we can:

- Compute KL term in ELBO in closed form (choosing $p(\mathbf{z})$ Gaussian)

$$\mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})\right] - \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \mid\mid p(\mathbf{z}))$$

- Sample $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ using the reparametrization trick

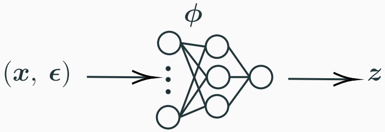


to compute gradients

$$\nabla_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})\right] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\phi}} \log p_{\boldsymbol{\theta}}(\mathbf{x} \mid z_{\boldsymbol{\phi}}(\mathbf{x}, \boldsymbol{\epsilon})\right]$$

Implicit Approximate Posterior (II)

To increase VAE flexibility, we want $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ to be implicit.
Beforehand $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ was a Gaussian and it was only parametrized by a NN.
Now define $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ as a black-box NN from which we can sample:



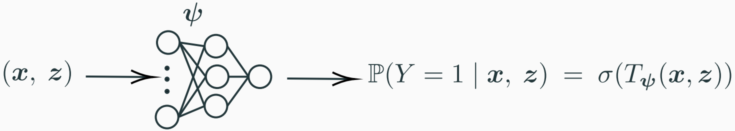
Due to its implicit form, we can't easily compute the KL term in the ELBO

$$\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \mid\mid p(\mathbf{z})).$$

For this reason, we introduce a new neural network such that, after learning is complete, it will approximate this KL term.

Implicit Approximate Posterior (III)

Discriminator network $\sigma(T_{\boldsymbol{\psi}}(\mathbf{x},\mathbf{z}))$ is trained to classify $Y=1$ when $(\mathbf{x},\mathbf{z}) \sim p_{\text{data}}(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ and $Y=0$ when $(\mathbf{x},\mathbf{z}) \sim p_{\text{data}}(\mathbf{x})p(\mathbf{z})$.



It learns $\boldsymbol{\psi}^*$ by maximizing Binary Cross Entropy

$$\max_{\boldsymbol{\psi}} \mathbb{E}_{p_{\text{data}}(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log \sigma(T_{\boldsymbol{\psi}}(\mathbf{x},\mathbf{z}))\right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x})p(\mathbf{z})}\left[\log(1-\sigma(T_{\boldsymbol{\psi}}(\mathbf{x},\mathbf{z}))\right]$$

Since (thanks to Bayes Rule) we have

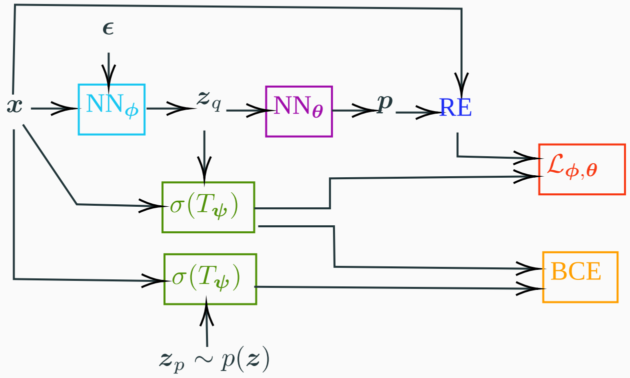
$$\mathbb{P}(Y=1 \mid \mathbf{x},\mathbf{z}) = \sigma\left(\log q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) - \log p(\mathbf{z})\right)$$

We can approximate the KL term with the discriminator output

$$\mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) - T_{\boldsymbol{\psi}^*}(\mathbf{x},\mathbf{z})]$$

Visualizing AVB

Encoder is now implicit and outputs $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ directly. Discriminator distinguishes between $(\mathbf{x}, \mathbf{z}) \sim p_{\text{data}}(\mathbf{x})q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ and $(\mathbf{x}, \mathbf{z}) \sim p_{\text{data}}(\mathbf{x})p(\mathbf{z})$. It is trained via Binary Cross Entropy. Decoder takes encoder output directly and Reconstruction Error works as in VAE. ELBO takes RE as usual but KL replaced by discriminator output before sigmoid.



AVB with Adaptive Contrast

Since $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ and $p(\mathbf{z})$ will be very different, the discriminator might struggle. Introduce auxiliary distribution $r_\alpha(\mathbf{z} \mid \mathbf{x}) \approx q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$

$$\begin{aligned}\mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x}) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) + \log p(\mathbf{z})] \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log r_\alpha(\mathbf{z} \mid \mathbf{x})] - \mathrm{KL}(q_{\boldsymbol{\phi}} \mid\mid r_\alpha)\end{aligned}$$

Introduce a new discriminator network distinguishing samples $(\mathbf{x}, \mathbf{z}) \sim p_{\text{data}}(\mathbf{x}) q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ from $(\mathbf{x}, \mathbf{x}) \sim p_{\text{data}}(\mathbf{x}) r_\alpha(\mathbf{z} \mid \mathbf{x})$ as before.

$$\mathcal{L}_{\theta,\boldsymbol{\phi}}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log r_\alpha(\mathbf{z} \mid \mathbf{x}) - T_{\boldsymbol{\psi}^*}(\mathbf{x}, \mathbf{z})\right]$$

Experiments

- AVB captures multi-modality.
- Adaptive Contrast makes learning more robust and improves posterior quality.

Thank you

References i