Deepmind - Modern LVM

Saturday, 27 February 2021 11:48

- Generative models are probabilistic models of high-dimensional data.
  - Describe the probabilistic process of generating an observation.
  - Emphasize on capturing dependence between the dimensions.
  - Provide a way of generating new data points.
- Types of generative models:
  1. Autoregressive models:
     - RNN & transformer language models, NADE, Pixel CNN, WaveNet
  2. Latent variable models:
     - Tractable: Invertible / flow-based models (RealNVP, Glow ..)
     - Intractable: Variational autoencoders
  3. Implicit models:
     - Generative Adversarial Networks (GANs)

- Auto-regressive models:
  - Model the 1-dimensional conditional distributions instead of modeling the joint distribution directly;
    - Based on the chain rule $p(x) = \prod_{d=1}^{D} p(x_d | x_1, \dots, x_{d-1})$
    - Trained with maximum likelihood.
  - Pros: 1. 1-dimensional dist. are easy to model.
    2. Simple and efficient training.
    3. No sampling at training time.
  - Cons: 1. Slow, sequential generation (one dimension at a time)
    2. Usually much better at modeling local structure than global structure.

- Latent variable models:
  - Specify the generative process in terms of unobserved / latent variables and the transformation that maps them to the observation.
    - Trained with maximum likelihood (usually with approximations)
  - Pros: 1. Powerful and well-understood framework.
    2. Easy to incorporate prior knowledge / structure into models well-suited for representation learning / interpretability
    3. Fast generation.
  - Cons: 1. Conceptually more complex than fully observed models.
    2. Need to use approximate inference or restricted models.

- Generative Adversarial Networks:
  - Model: NN that maps noise vectors to observations.
  - Training: use the learning signal from a classifier trained to discreminate between samples from the model and the training data.
  - Pros: 1. Can generate very realistic images.
    2. Conceptually simple implementation.
    3. Fast generation.
  - Cons: 1. Cannot be used to compute probability of observations.
    2. "Mode collapse": Models ignore regions of the data dist.
    3. Training can be unstable and require many tricks to work.

- Latent variable models: define a distribution over observations x by using a (vector) latent variable z and specifying:
  - The prior distribution $p(z)$ for the latent variable.
  - The likelihood $p(x|z)$ connects the latent variable to the observation.
  - The prior and the likelihood define the joint distribution $p(x,z) = p(x|z)\, p(z)$
  - We need the marginal likelihood $p(x)$ and the posterior $p(z|x)$
  - We can think of latent variable value as an explanation for the observation.
  - To generate an observation from the model: $z \sim p(z) \Rightarrow x \sim p(x|z)$
  - Inference: Going from observation to latent values.

    i.e. computing the posterior distribution for the given observation:

    $$p(z|x) = \frac{p(x,z)}{p(x)} = \frac{p(x,z)}{\int p(x,z)\, dz}$$

    This requires computing the marginal likelihood: $p(x) = \int p(x,z)\, dz$
  - Generating pairs (x,z): $z \sim p(z) \Rightarrow x \sim p(x|z)$

    or $x \sim p(x) \Rightarrow z \sim p(z|x)$
  - Maximizing likelihood: $\theta^{ML} = \arg\max_{\theta} \sum_{i=1}^{N} \log p_\theta(x_i)$
  - The gradient for a single point:

    $$\nabla_\theta \log p_\theta(x) = \int p_\theta(z|x)\, \nabla_\theta \log p_\theta(x,z)\, dz$$

- Invertible models (aka. normalizing flows):
  - Approximate the data distribution by transforming the prior distribution using an invertible function; can be done (1) jointly: for all dimensions or autoregressively: one dimension at a time.
  - Generate observations by applying an invertible and differentiable tranformation $f_\theta(z)$ to sample from the prior: $z \sim p(z) \Rightarrow x = f_\theta(z)$

  The marginal likelihood: we can relate densities of x and z:

  $$p_\theta(x) = p_\theta(z) \left| \det \frac{\partial z}{\partial x} \right| = p_\theta(f_\theta^{-1}(x)) \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right|$$

  change of variables formula

  - The deteminant of the Jacobian takes into account the local chang in folume when applying the transformation.
  - Independent Component Analysis (ICA):

    uses fractional prior and a linear mapping (a square matrix):

    $$z \sim \prod_i p(z_i) \Rightarrow x = Az$$
  - Limitations: 1. The latent space and data space dimensionality must be the same.
    2. The latent variable have to be continuous.
    3. Observations have to be continuous or quantized.
    4. Expressive models require a lot of layers, parameters.
    5. Lack of flexibility in model design.

- Solution: use approximate inference for intractable models.
- Approximate inference:
  1. Markov Chain Monte Carlo: generate samples from the exact posterior using a Markov Chain:
     - +: very general; exact in the limit of infinite time/computation.
     - -: computationally expensive; convergence is hard to diagnose.
  2. Variational inference: approximate the posterior with a tractable distribution; e.g. fully factorized or auto-regressive.
     - +: fairly efficient, inference is reduced to optimization w.r.t. dist. parameters.
     - -: cannot trade computation for accuracy easily.

- Variational inference:
  - We approximate the exact posterior $p_\theta(z|x)$ with a variational posterior $q_\phi(z|x)$
  - We can use any distribution for $q_\phi(z|x)$ as long as:
    1. We can sample from it.
    2. We can compute $\log q_\phi(z|x)$ and its gradient w.r.t. ϕ

  Training: The variational posterior induces a variational lower bound $\mathcal{L}_{\theta,\phi}(x)$ on the marginal log-likelihood.

  → we train the model by maximizing $\mathcal{L}_{\theta,\phi}(x)$ w.r.t both model parameters θ and the variational parameter ϕ.

  Bounding the marginal log-likelihood:

  $$\log p_\theta(x) = \log \int p_\theta(x,z)\, dz = \log \int q_\phi(z) \frac{p_\theta(x,z)}{q_\phi(z)}\, dz \geq \int q_\phi(z) \log \frac{p_\theta(x,z)}{q_\phi}\, dz$$

  Jensen's inequality

  $$\Rightarrow \mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z)}\right] \leq \log p_\theta(x)$$

  - Evidence Lower Bound (ELBO):

    Is obtained by using the variational posterior $q_\phi(z|x)$ as $q_\phi(z)$

    $$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right]$$
  - Importance-Weighted Bound (IWAE)
    - A multi-sample generalization of ELBO;
    - Allows to control the tightness of the bound; the accuracy of approximation by increasing the number of samples.
  - Rewriting the ELBO:

    $$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x)\, p_\theta(z|x)}{q_\phi(z|x)}\right]$$

    $$= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x)\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(z|x)}{q_\phi(z|x)}\right]$$

    doesn't depend on z

    $$= \log p_\theta(x) - KL(q_\phi(z|x) \,||\, p_\theta(z|x))$$

    Variational gap

    → the ELBO is difference between two intractable quantities but it is tractable.
  - Variational pruning: when the pressure fro VI to make the exact posterior be closer to the variational posterior often result in some of the latent variables not being used by the model. (posterior collapse for VAE).

- Problem: the posterior distribution is different for each observation
- Amortized variational inference:

  Instead of performing optimization for each datapoint, we can train a NN; called the inference network, to predict the variational parameters from the observation.

  → ϕ in $q_\phi(z|x)$ now refers to the inference network parameters.

  → the inference network is trained jointly with the model by maximizing the ELBO.

- Gradient w.r.t. model parameters:

  using Monte Carlo sampling from the variational posterior:

  $$\nabla_\theta \mathcal{L}_{\theta,\phi}(x) = \nabla_\theta \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] = \mathbb{E}_{q_\phi(z|x)}\left[\nabla_\theta \log p_\theta(x,z)\right]$$

  $$\approx \frac{1}{k} \sum_{k=1}^{k} \nabla_\theta \log p_\theta(x, z^k) \quad ; \quad z^k \sim q_\phi(z|x)$$

- Gradient w.r.t. variational parameters:

  Two major types of gradient estimator $\nabla_\phi \mathbb{E}_{q_\phi(z)}[f(z)]$:
  1. REINFORCE / likelihood-ratio estimator
     - +: very general (discrete & continuous, non-differentiable $f(z)$)
     - -: high variance.
  2. Reparametrization / path-wise estimator
     - -: less general (continuous, differentiable $f(z)$)
     - +: relatively low variance.
  - Reparametrization trick:

    we reparametrize a sample from $q_\phi(z)$ by expressing it as a function of a sample ϵ from some fixed distribution $p(\epsilon)$:

    $$z \sim g(\epsilon, \phi)$$

    $$\Rightarrow \nabla_\phi \mathbb{E}_{q_\phi(z)}[f(z)] = \nabla_\phi \mathbb{E}_{p(\epsilon)}[f(g(\epsilon,\phi))]$$

    $p(\epsilon)$ doesn't depend on ϕ: $= \mathbb{E}_{p(\epsilon)}[\nabla_\phi f(g(\epsilon,\phi))]$

    chain rule: $= \mathbb{E}_{p(\epsilon)}[\nabla_z f(z)|_{z=g(\epsilon,\phi)} \nabla_\phi g(\epsilon,\phi)]$

    Reparametrizing a Gaussian variable $z \sim \mathcal{N}(\mu, \sigma^2)$

    $$z = \mu + \sigma\epsilon \quad ; \quad \epsilon \sim \mathcal{N}(0, 1^2)$$

- Variational autoencoder: a generative model with continuous latent variables:
  - The likelihood $p_\theta(x|z)$ and the variational posterior $q_\phi(z|x)$ are NN.
  - Both prior and the variational posterior are usually fully factorized Gaussians.
  - VAEs are trained using amortized variational inference, taking advantage of the reparametrization trick.
  - prior: $p(z) = \mathcal{N}(0,1)$
  - likelihood / decoder:

    for binary data: $p_\theta(x|z) = \text{Bernoulli}(NN_\theta(z))$

    for real-valued data: $p_\theta(x|z) = \mathcal{N}(NN_\theta(z), \text{diag}(NN_\theta(z)))$
  - Variational posterior / encoder:

    $$q_\phi(z|x) = \mathcal{N}(NN_\phi(x), \text{diag}(NN_\phi(x)))$$
  - trained using by maximizing the ELBO.