Auto-Encoding Variational Bayes

1 Citation

Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

`https://arxiv.org/pdf/1312.6114.pdf`

2 Abstract

By estimating an intractable posterior distribution with an approximate inference model (trained with SGD), we can make inferences in directed graphical models with continuous latent variables, intractable posteriors, and larger datasets.

3 Introduction

Variational Bayes (VB) is used to approximate intractable posteriors, but can sometimes itself be intractable. With a clever reparameterization, we propose a Stochastic Gradient VB (SGVB) estimator that can approximate the posterior. For an i.i.d dataset with continuous latent variables, we use the Auto-Encoding VB (AEVB) model, which can be used as a SGVB estimator. When this model is a neural network, we call it a variational autoencoder.

4 Method

Suppose we have a dataset generated by (1) generating latent variables $\boldsymbol{z^{(i)}}$ from a prior $p_{\theta^*}(\boldsymbol{z})$ and (2) the data point is generated with $p_{\theta^*}(\boldsymbol{x}|\boldsymbol{z})$. We assume that the posterior $p(\boldsymbol{z}|\boldsymbol{x})$ is intractable and the dataset is large (so Markov Chain Monte Carlo won't work).

We aim to (1) infer $\boldsymbol{\theta}$ from the data, (2) infer $\boldsymbol{z}$ from $\boldsymbol{\theta}$ and $\boldsymbol{x}$, and (3) marginalize over $\boldsymbol{x}$. We approximate the posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ with a recognition network (also called an encoder) $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. We observe that

$$\log p_\theta(\boldsymbol{x}^{(i)}) = D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_\theta(\boldsymbol{z}|\boldsymbol{x^{(i)}})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) \tag{1}$$

The second term, the variational lower bound, is $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) = -D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_\theta(\boldsymbol{z})) + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})}[\log p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z})]$

We further parameterize $\tilde{\boldsymbol{z}} = g_\theta(\boldsymbol{\epsilon}, \boldsymbol{x})$ where $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ (this enables easier sampling). After some other simplifications, we can estimate the variational lower bound with:

$$\tilde{\mathcal{L}}^B(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) = -D_{KL}(q_\theta(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_\theta(\boldsymbol{z})) + \frac{1}{L}\sum_{l=1}^{L} \log(p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z}^{(i,l)})) \tag{2}$$

Given a minibatch of size M, we can estimate this as $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}) \approx \tilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{X}^M) = \frac{N}{M} \sum_{i=1}^{M} \tilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{x}^{(i)})$

We set $M = 100$ and $L = 1$. In the approximation $\tilde{\mathcal{L}}$, the KL divergence acts as a regularizer pushing the recognition function towards the prior. The recognition function encodes noise to produce $\boldsymbol{z}$, which is then reconstructed with the likelihood function to create $\boldsymbol{x}^{(i)}$.

Notice our reparameterization trick where we have defined $p(\epsilon)$ as some distribution that we are allowed to choose. We can make this distribution something that is easy to sample from.

5 Example: Variational Auto-Encoder

We'll use a neural network to approximate $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. We assume $p_\theta(\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$. We assume $p_\theta(\boldsymbol{x}|\boldsymbol{z})$s is a multivariate Gaussian (or Bernoulli if the data is binary) where the distribution parameters are estimated from a multi-layer perceptron (MLP) parameterized by $\boldsymbol{\theta}$ and operating on $\boldsymbol{z}$. We further assume that

$$\log(q_\theta(\boldsymbol{z}|\boldsymbol{x}^{(i)})) \sim \log(\mathcal{N}(\boldsymbol{z}; \mu^{(i)}, \boldsymbol{\sigma}^{(i)}\boldsymbol{I})) \tag{3}$$

where $\boldsymbol{\mu}^{(i)}$ and $\boldsymbol{\sigma}^{(i)}$ are computed from an MLP parameterized by $\boldsymbol{\phi}$ and operating on $\boldsymbol{x}^{(i)}$. We let $\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(0, \boldsymbol{I})$ and we thus compute:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}^{(i)}) \approx \frac{1}{2} \sum_{j=1}^{J} (1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2) + \frac{1}{L} \sum_{l=1}^{L} \log(p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z}^{(i,l)})) \tag{4}$$

$$\boldsymbol{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)} \tag{5}$$

$$\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(0, \boldsymbol{I}) \tag{6}$$

6 Related Work

The wake-sleep function also approximates the recognition function, but requires two objective functions. There are techniques to regularize autoencoders. Generative Stochastic Networks use noisy autoencoders to learn transition operator of a Markov chain.

7 Experiments

We train on MNIST and Frey Faces. MNIST had Bernoulli output units while Frey Faces had Gaussian (constrained to range [0, 1] with a sigmoid unit) output units. We train with Adagrad with weight decay. Weights were initialized from a zero-mean Gaussian. We compare against wake-sleep.

We trained generative models (decoders) and recognition models (encoders) for the likelihood lower bound. We also did this for the marginal likelihood.

8 Conclusion

We show how to efficiently approximate the variational lower bound using a neural network trained with SGD.

9 Future Work

We can learn hierarchical generative architectures, time series models, and apply this to the global parameters of a directed graphical model.