Project report
Improving Explorability in Variational Inference with Annealed Variational Objectives.

Mikhail Kurenkov, Timur Chikichev, Aleksei Pronkin *
pronkinalexeyviktorovich@gmail.com *
https://github.com/alexey-pronkin/annealed

Abstract

We represent the results of paper [**?**]. The main work of research is a special procedure, applied into the training of hierarchical variational methods. The method called Annealed Variational Objectives (AVO) has to solve the problem of limited posterior distribution density. The method facilitates learning by integrating energy tempering into the optimization objective.

The paper presents contains experiments on the proposed method. These experiments represent the drawbacks of biasing the true posterior to be unimodal, and show how proposed method solve this problem. We repeat experiments from [**?**] and compare performance of AVO with normalizing flows (NF) and variational auto-encoders (VAE). Additionally we make experiments with deterministic warm up (analogously to AVO), which when applied to NF and VAE may benefits in better space exploration.

Introduction

With variational inference, we have some variational distribution we may use to generate samples. The resulting variance can be lowered by two ways, increasing number of samples and increasing approximation accuracy. If variational inference has bad uncertainty approximation (Turner and Sahani (2011)), we will receive bias in statistics in terms of overconfidence and inaccuracy. The statistics we check in models are marginal likelihood of data and the predictive posterior. The same in the amortized VI setups, the representation of data will require better exploration from approximation model during training.

To express the bias induced by a non-rich and non-expressive variational family, the objective can be written as KL-divergence between proposal and target distributions.

Variational inference objective: $\mathrm{F(q)} = \mathrm{E}_q[\log q(z) - \log f(z)] = D_{KL}(q||f)$

Due to KL-divergence, the resulting approximation will have low probability mass in regions with low density. The variational approximation may escape points with sufficient statistics in true target, but with small local density. For multi-modal target distributions, not all target space will be covered and the model will loose some sufficient statistics.

Annealing techniques may increase exploration of the target density.

Alpha-annealing (expressiveness):

$\mathrm{E}_q[\log q(z) - \alpha \log f(z)]$

where $\alpha \sim \frac{1}{T}$, and T is temperature which defines the speed of approximate model changes, e.g. learning rate. When α goes from zero to 1, we obtain the usual objective, but with full energy landscape covered.

The problem, with low penalty on the energy term, the whole procedure is time consuming. This is because multiple inferences are required on each maximization step (deep neural networks, hierarchical models, etc.).

Beta-annealing (optimization):

Deterministic warm-up (Raiko et al., 2007) is applied to improve training of a generative model. p(x, z) = p(x—z)p(z)

The joint likelihood is equal to the un-normalized true posterior $f(z) = p(z|x)$.

The annealed objective is (negative ELBO): $\mathrm{E}_q[\beta(\log q(z) - \log p(z)) - \log p(x|z)]$

In annealed objective, the β is annealed from 0 to 1. This disables the regularisation of posterior to be like a prior. First training the negative log-likelihood, we train the decoder independently. With this the model is trained to fit the data, so we have more deterministic auto-encoder. With this approach we additionally lose in latent space exploration.

The model: Latent variable model with a joint probability $p_\theta(x, z) = p(x|z)p(z)$.

x and z are observed and latent variables, θ - model parameters to be learned.

Training procedure, given expected complete data log likelihood over q:

$\max_\theta E_{q(z)}[\log p_\theta(x, z)]$

Conditional $q(z|x)$

1. tractable: Expectation-Maximization(EM) algorithm.

2. non-tractable: approximate the true posterior (MCMC, VI)

Variational distribution subfamilies with expressive parametric form

1. Hierarchical Variational Inference(HVI)

2. Auxiliary variable methods

3. Normalizing flows

In HVI, we use a latent variable model $q(z_T) = \int q(z_T, z_{t<T})dz_{t<T}$, where $t < T$ denoting latent variables.

We use reverse network $r(z_{t<T})$ to lower bound intractable $q(z_T)$.

- $\mathrm{E}_{q(z_T)}[\log q(z_T)] \geq -E_{q(z_T)}[\ \log q(z_T) + D_{KL}(q(z_{t<T}|z_T)||r(z_{t<T}|z_T))] =$
$=$ -$\mathrm{E}_{q(z_T,z_{t<T})}[\ \log q(z_T|z_{t<T})q(z_{t<T}) - \log r(z_{t<T}|z_T)]$

The variational lower bound is: L(x) $=$
$\mathrm{E}_{q(z_T,z_{t<T})}[\ \log \frac{p(x,z_T)r(z_{t<T}|z_T)}{}q(z_T|z_{t<T})q(z_{t<T})]$

As the $q(z)$ is one from chosen distribution subfamilies, we have the capability to represent any posterior distribution. If possible to invert $q(z_T|z_{t<T})$, we choose r to be its invert transformation. This is the so called inverse auto-regressive flow. The KL term is zero, the variance is lower, the entropy is computed via change of the volume formula.

$$q(z_T) = q(z_{T-1})|\frac{\partial z_T}{\partial z_{T-1}}|^{-1} \tag{1}$$

Loss function tempering: annealed importance sampling

Annealed importance sampling(AIS) is an MCMC method with same concept as alpha annealing, it let the variational distribution be more exploratory early on during training.

We have an extended state space with $z_0, .., .z_T$ latent variables. z_0 is sampled from simple distribution (Gaussian normal prior distribution $p(z)$). Particles are sequentially sampled from the transition operators $q_t(z_t|z_{t1})$.

To define transition operators, we design a set of intermediate target densities as $\tilde{f}_t = f_T^{\tilde{\alpha}_t} f_T^{1\tilde{-}\alpha_t}$. This is the set of targets defined as a mixture of initial (normal) and target (complex multi-modal) distributions.

For intermediate targets to be invariant, the transitions are constructed as Markov chain with the following weights: $\mathrm{w}_j = \frac{\tilde{f}_1(z_1)\tilde{f}_2(z_2)}{\tilde{f}_0(z_1)\tilde{f}_1(z_2)} ... \frac{\tilde{f}_T(z_T)}{f_{T\tilde{-}1}(z_T)}$

For the estimate to be accurate we need a long sequence of transitions, computationally difficult.

Annealed Variational Objectives(AVO)

Similar to AIS and alpha-annealing, authors of [?] propose to integrate energy tempering into the optimization objective of the variational distribution.

As in AIS, we consider an extended state space with same transitional targets. The marginal $q_T(z_T)$ is an approximate posterior. To define incremental steps, we construct T forward transition operators and T backward operators. We construct intermediate targets as an interpolation between the true (unnormalized) posterior and the initial (normal) distribution: $\tilde{f}_t = f_T^{\tilde{\alpha}_t} f_T^{1\tilde{-}\alpha_t}$, where $\alpha \in [0, 1]$.

Different from AIS, we learn the parametric transition operators which are assigned to each transition pair. We have a sequence of one layer networks as a result.

Annealed Variational Objectives(AVO):

$\max_{q_t(z_t|z_{t-1})r_t(z_{t-1}|z_t)} E_{q_t(z_t|z_{t-1})q_{t-1}(z_{t-1})}[\ \log \frac{\tilde{f}_t(z_T)r_t(z_{t-1}|z_t)}{q_t(z_t|z_{t-1})q_{t-1}(z_{t-1})}]$

1 Experiments

VAE experiment on MNIST dataset

Same decoder and encoder,

2 hidden layers with dimension 300,

40 - latent space size,

LeakyReLU activation function,

Batch normalization.

Optimizer - Adam (lr - 1e-3), batch$_size = 64, epochs = 25$

HVI - 5 stochastic transition operators (hidden size - 40)

Beta annealing

Beta0 = 0.2

Gamma = 2e-4

2 Results

3 Conclusion

4 Resources

Github repository: `https://github.com/alexey-pronkin/annealed`

Presentation:

Acknowledgements

The project represents the paper [?]. We use [?] as an introduction to the problem and [?] as an introduction to generalized variational inference problem.

We use and rewrite some code from
`https://github.com/joelouismarino/iterative_inference/`,
`https://github.com/jmtomczak/vae_householder_flow`,
`https://github.com/AntixK/PyTorch-VAE`, ,
`https://github.com/haofuml/cyclical_annealing` and
`https://github.com/ajayjain/lmconv`.

We assume, that first two repositories were used in the original paper [**?**] closed source code.

We want to try to apply annealing strategies for some of SoTA AE for MNIST `https://paperswithcode.com/sota/image-generation-on-mnist` if we will have time.