Summer Review 10

Semi Amortized Variational Autoencoders

Yoon Kim[1], *SamWiseman*[1], *AndrewC.Miller*[1], *DavidSontag*[2] *andAlexanderM.Rush*[1]

ICML 2018

Paper Link

1: School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA

2: CSAIL IMES, Massachusetts Institute of Technology, Cambridge, MA, USA

Reviewed by : Arshdeep Sekhon

[1]Department of Computer Science, University of Virginia
`https://qdata.github.io/deep2Read/`

Introduction

- Variational inference (VI) is a framework for approximating an intractable distribution by optimizing over a family of tractable surrogates
- Traditional VI algorithms iterate over the observed data and update the variational parameters with closed-form coordinate ascent updates that exploit conditional conjugacy
- This style of optimization is challenging to extend to large datasets and non-conjugate models

Stochastic and Amortized Variational inference

- the variational parameters for each data point are randomly initialized and then optimized to maximize the evidence lower bound (ELBO) with, for example, gradient ascent
- These updates are based on a subset of the data, making it possible to scale the approach.
- amortized variational inference, the local variational parameters are instead predicted by an inference (or recognition) network
- which is shared (i.e. amortized) across the dataset.
- VAEs utilize AVI for inference and jointly train the generative model alongside the inference network.

SVI vs AVI

- (+) SVI gives good local (i.e. instance-specific) distributions within the variational family
- (-) but requires performing optimization for each data point
- AVI has fast inference,
- (-) but having the variational parameters be a parametric function of the input may be too strict of a restriction
- (-) hence,its parameters may be updated based on suboptimal variational parameters
- amortization gap (the gap between the log-likelihood and the ELBO due to amortization) can be significant for VAEs, especially on complex datasets

Background: Notation

- $f : R^m \rightarrow R$ scalar valued function
- partitioned inputs: $[u_1, \ldots, u_m]$
- $\Sigma_{i=1}^{m} dim(u_i) = n$

Background:Variational Inference

- $z \sim p(z)$
- $x \sim p(x|z, \theta)$
- ELBO(λ, θ, x): $logp(x; \theta) \geq E_{q(z;\lambda)}[logp(x|z)] - KL[q(z; \lambda)||p(z)]$
- given a dataset $x_1, \ldots, x_N$ and need to find variational parameters $\lambda_1, \ldots, \lambda_N$ and generative model parameters that jointly maximize ELBO

SVI

- Sample $x \sim p_D(x)$
- Randomly initialize λ_0
- For $k = 0, \ldots, K-1$
- $\lambda k + 1 = \lambda_k + \alpha \nabla_{lambda} ELBO(\lambda_k; \theta; x)$
- 4. Update θ based on $\nabla_\theta ELBO(\lambda_k; \theta; x)$
- because of this block coordinate ascent approach the variational parameters, λ are optimized separately from θ, potentially making it difficult for θ to adapt to local optima.

AVI

- Sample $x \sim p_D(x)$
- 2. Set $\lambda_0 = enc(x; \phi)$
- 3. Update θ based on $\nabla_\theta ELBO(\lambda_k; \theta; x)$ (which in this case is equal to the total derivative)
- 4. Update ϕ based on $\frac{d(ELBO(\lambda, \theta, x))}{d\phi} = \frac{d\lambda}{d\phi} \nabla_\phi ELBO(\lambda, \theta, x)$
- The inference network is learned jointly alongside the generative model with the same loss function, allowing the pair to coadapt.
- requiring the variational parameters to be a parametric function of the input may be too strict of a restriction and can lead to an amortization gap. This gap can propagate forward to hinder the learning of the generative model if θ is updated based on suboptimal λ. While we describe the various algorithms for a specific data point, in practice we use mini-batches.

Semi-Amortized Variational Autoencoders

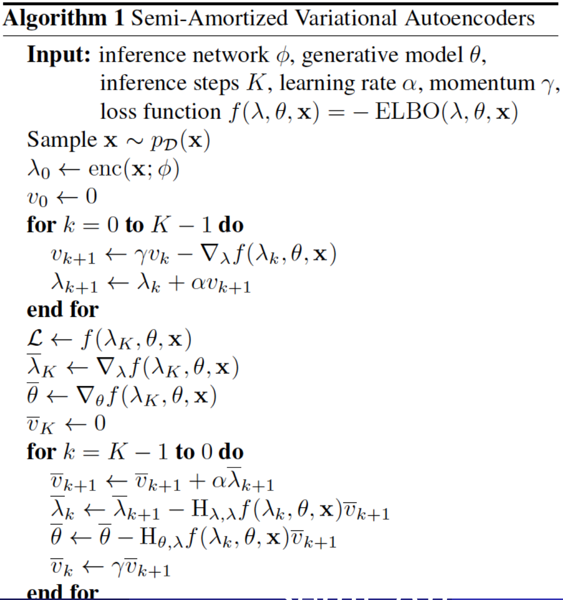
- utilize an inference network over the input to give the initial variational parameters
- subsequently run SVI to refine them.
-
- Sample $x \sim p_D(x)$
- 2. Set $\lambda_0 = enc(x; \phi)$
- 3. For $k = 0, \ldots, K-1$, set $\lambda_{k+1} = \lambda_k + \alpha \nabla_{lambda} ELBO(\lambda_k; \theta; x)$
- 4. Update θ based on $\dfrac{d(ELBO(\lambda_k; \theta; x))}{d(\theta)}$
- 5. Update ϕ based on $\dfrac{d(ELBO(\lambda_K; \theta; x))}{d(\phi)}$

savae chain rule

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\lambda_0} = \frac{\mathrm{d}\lambda_1}{\mathrm{d}\lambda_0}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\lambda_1} = (\mathbf{I} + \alpha \mathrm{H}_{\lambda,\lambda}\,\mathrm{ELBO}(\lambda_0, \theta, \mathbf{x}))\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\lambda_1}$$

$$= \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\lambda_1} + \alpha \mathrm{H}_{\lambda,\lambda}\,\mathrm{ELBO}(\lambda_0, \theta, \mathbf{x})\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\lambda_1}$$

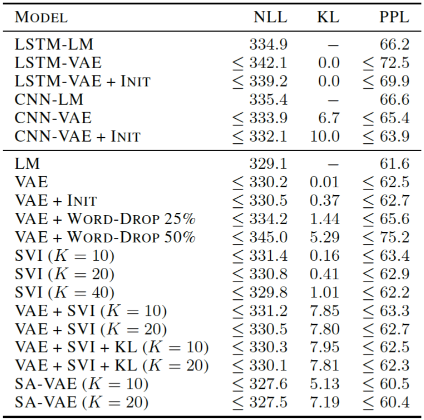
SVAE Algorithm



Algorithm 1 Semi-Amortized Variational Autoencoders

```
Input: inference network $\phi$, generative model $\theta$,
       inference steps $K$, learning rate $\alpha$, momentum $\gamma$,
       loss function $f(\lambda, \theta, \mathbf{x}) = -\,\mathrm{ELBO}(\lambda, \theta, \mathbf{x})$
Sample $\mathbf{x} \sim p_{\mathcal{D}}(\mathbf{x})$
$\lambda_0 \leftarrow \mathrm{enc}(\mathbf{x}; \phi)$
$v_0 \leftarrow 0$
for $k = 0$ to $K - 1$ do
    $v_{k+1} \leftarrow \gamma v_k - \nabla_\lambda f(\lambda_k, \theta, \mathbf{x})$
    $\lambda_{k+1} \leftarrow \lambda_k + \alpha v_{k+1}$
end for
$\mathcal{L} \leftarrow f(\lambda_K, \theta, \mathbf{x})$
$\overline{\lambda}_K \leftarrow \nabla_\lambda f(\lambda_K, \theta, \mathbf{x})$
$\overline{\theta} \leftarrow \nabla_\theta f(\lambda_K, \theta, \mathbf{x})$
$\overline{v}_K \leftarrow 0$
for $k = K - 1$ to $0$ do
    $\overline{v}_{k+1} \leftarrow \overline{v}_{k+1} + \alpha \overline{\lambda}_{k+1}$
    $\overline{\lambda}_k \leftarrow \overline{\lambda}_{k+1} - \mathrm{H}_{\lambda,\lambda} f(\lambda_k, \theta, \mathbf{x}) \overline{v}_{k+1}$
    $\overline{\theta} \leftarrow \overline{\theta} - \mathrm{H}_{\theta,\lambda} f(\lambda_k, \theta, \mathbf{x}) \overline{v}_{k+1}$
    $\overline{v}_k \leftarrow \gamma \overline{v}_{k+1}$
end for
```

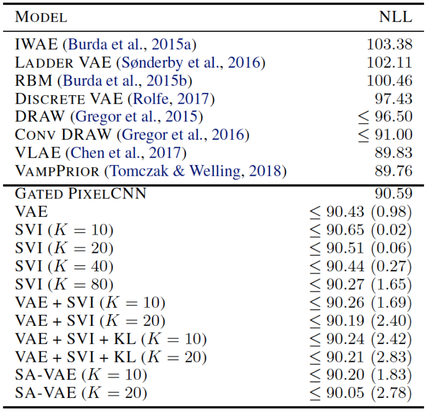
Results



Model	NLL	KL	PPL
LSTM-LM	334.9	–	66.2
LSTM-VAE	$\leq$ 342.1	0.0	$\leq$ 72.5
LSTM-VAE + Init	$\leq$ 339.2	0.0	$\leq$ 69.9
CNN-LM	335.4	–	66.6
CNN-VAE	$\leq$ 333.9	6.7	$\leq$ 65.4
CNN-VAE + Init	$\leq$ 332.1	10.0	$\leq$ 63.9
LM	329.1	–	61.6
VAE	$\leq$ 330.2	0.01	$\leq$ 62.5
VAE + Init	$\leq$ 330.5	0.37	$\leq$ 62.7
VAE + Word-Drop 25%	$\leq$ 334.2	1.44	$\leq$ 65.6
VAE + Word-Drop 50%	$\leq$ 345.0	5.29	$\leq$ 75.2
SVI ($K = 10$)	$\leq$ 331.4	0.16	$\leq$ 63.4
SVI ($K = 20$)	$\leq$ 330.8	0.41	$\leq$ 62.9
SVI ($K = 40$)	$\leq$ 329.8	1.01	$\leq$ 62.2
VAE + SVI ($K = 10$)	$\leq$ 331.2	7.85	$\leq$ 63.3
VAE + SVI ($K = 20$)	$\leq$ 330.5	7.80	$\leq$ 62.7
VAE + SVI + KL ($K = 10$)	$\leq$ 330.3	7.95	$\leq$ 62.5
VAE + SVI + KL ($K = 20$)	$\leq$ 330.1	7.81	$\leq$ 62.3
SA-VAE ($K = 10$)	$\leq$ 327.6	5.13	$\leq$ 60.5
SA-VAE ($K = 20$)	$\leq$ 327.5	7.19	$\leq$ 60.4

Table 2. Results on text modeling on the Yahoo dataset. Top results are from Yang et al. (2017), while the bottom results are from this work (+ Init means the encoder is initialized with a pretrained language model, while models with + Word-Drop are trained with word-dropout). NLL/KL numbers are averaged across examples, and PPL refers to perplexity. K refers to the number of inference steps used for training/testing.

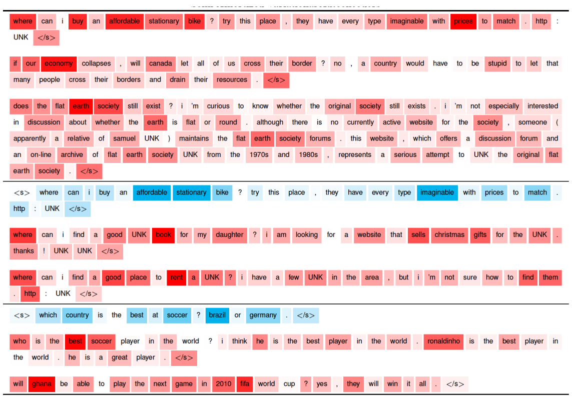
Results



Model	NLL
IWAE (Burda et al., 2015a)	103.38
Ladder VAE (Sønderby et al., 2016)	102.11
RBM (Burda et al., 2015b)	100.46
Discrete VAE (Rolfe, 2017)	97.43
DRAW (Gregor et al., 2015)	$\leq$ 96.50
Conv DRAW (Gregor et al., 2016)	$\leq$ 91.00
VLAE (Chen et al., 2017)	89.83
VampPrior (Tomczak & Welling, 2018)	89.76
Gated PixelCNN	90.59
VAE	$\leq$ 90.43 (0.98)
SVI ($K = 10$)	$\leq$ 90.65 (0.02)
SVI ($K = 20$)	$\leq$ 90.51 (0.06)
SVI ($K = 40$)	$\leq$ 90.44 (0.27)
SVI ($K = 80$)	$\leq$ 90.27 (1.65)
VAE + SVI ($K = 10$)	$\leq$ 90.26 (1.69)
VAE + SVI ($K = 20$)	$\leq$ 90.19 (2.40)
VAE + SVI + KL ($K = 10$)	$\leq$ 90.24 (2.42)
VAE + SVI + KL ($K = 20$)	$\leq$ 90.21 (2.83)
SA-VAE ($K = 10$)	$\leq$ 90.20 (1.83)
SA-VAE ($K = 20$)	$\leq$ 90.05 (2.78)

Table 3. Results on image modeling on the OMNIGLOT dataset. Top results are from prior works, while the bottom results are from this work. Gated PixelCNN is our autoregressive baseline, and K refers to the number of inference steps during training/testing. For the variational models the KL portion of the ELBO is shown in parentheses.

Results



where can i buy an affordable stationary bike ? try this place , they have every type imaginable with prices to match . http : UNK </s>

if our economy collapses , will canada let all of us cross their border ? no , a country would have to be stupid to let that many people cross their borders and drain their resources . </s>

does the flat earth society still exist ? i 'm curious to know whether the original society still exists . i 'm not especially interested in discussion about whether the earth is flat or round . although there is no currently active website for the society , someone (apparently a relative of samuel UNK) maintains the flat earth society forums . this website , which offers a discussion forum and an on-line archive of flat earth society UNK from the 1970s and 1980s , represents a serious attempt to UNK the original flat earth society . </s>

<s> where can i buy an affordable stationary bike ? try this place , they have every type imaginable with prices to match . http : UNK </s>

where can i find a good UNK book for my daughter ? i am looking for a website that sells christmas gifts for the UNK . thanks ! UNK UNK </s>

where can i find a good place to rent a UNK ? i have a few UNK in the area , but i 'm not sure how to find them . http : UNK </s>

<s> which country is the best at soccer ? brazil or germany . </s>

who is the best soccer player in the world ? i think he is the best player in the world . ronaldinho is the best player in the world . he is a great player . </s>

will ghana be able to play the next game in 2010 fifa world cup ? yes , they will win it all . </s>

Figure 3. (Top) Saliency visualization of some examples from the test set. Here the saliency values are rescaled to be between 0-100 within each example for easier visualization. Red indicates higher saliency values. (Middle) Input saliency of the first test example from the top (in blue), in addition to two sample outputs generated from the variational posterior (with their saliency values in red). (Bottom) Same as the middle except we use a made-up example. Best viewed in color.