Semi-Amortized Variational Autoencoders

Yoon Kim Sam Wiseman Andrew Miller
David Sontag Alexander Rush

Code: `https://github.com/harvardnlp/sa-vae`

Background: Variational Autoencoders (VAE) (Kingma et al. 2013)

Generative model:

- Draw $\mathbf{z}$ from a simple prior: $\mathbf{z} \sim p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$
- Likelihood parameterized with a deep model θ, i.e. $\mathbf{x} \sim p_\theta(\mathbf{x} \,|\, \mathbf{z})$

Training:

- Introduce variational family $q_\lambda(\mathbf{z})$ with parameters λ
- Maximize the evidence lower bound (ELBO)

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\lambda(\mathbf{z})}\Big[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\lambda(\mathbf{z})}\Big]$$

- VAE: λ output from an inference network ϕ

$$\lambda = \text{enc}_\phi(\mathbf{x})$$

Background: Variational Autoencoders (VAE) (Kingma et al. 2013)

Generative model:

- Draw $\mathbf{z}$ from a simple prior: $\mathbf{z} \sim p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$
- Likelihood parameterized with a deep model θ, i.e. $\mathbf{x} \sim p_\theta(\mathbf{x} \,|\, \mathbf{z})$

Training:

- Introduce variational family $q_\lambda(\mathbf{z})$ with parameters λ
- Maximize the evidence lower bound (ELBO)

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\lambda(\mathbf{z})}\Big[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\lambda(\mathbf{z})}\Big]$$

- VAE: λ output from an inference network ϕ

$$\lambda = \mathsf{enc}_\phi(\mathbf{x})$$

Background: Variational Autoencoders (VAE) (Kingma et al. 2013)

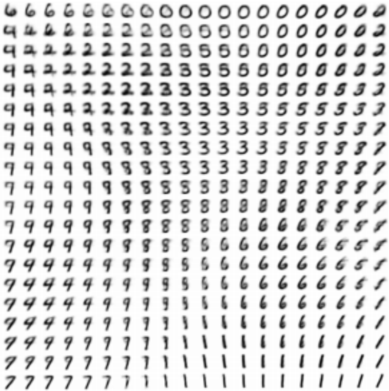
- **Amortized Inference**: *local* per-instance variational parameters $\lambda^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$ predicted from a *global* inference network (cf. per-instance optimization for traditional VI)
- **End-to-end**: generative model θ and inference network ϕ trained together (cf. coordinate ascent-style training for traditional VI)

Background: Variational Autoencoders (VAE) (Kingma et al. 2013)

- **Amortized Inference**: *local* per-instance variational parameters $\lambda^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$ predicted from a *global* inference network (cf. per-instance optimization for traditional VI)
- **End-to-end**: generative model θ and inference network ϕ trained together (cf. coordinate ascent-style training for traditional VI)

Background: Variational Autoencoders (VAE) (Kingma et al. 2013)

- Generative model: $\int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$ gives good likelihoods/samples
- Representation learning: $\mathbf{z}$ captures high-level features



VAE Issues: Posterior Collapse (Bowman al. 2016)

(1) Posterior collapse

- If generative model $p_\theta(\mathbf{x} \mid \mathbf{z})$ is too flexible (e.g. PixelCNN, LSTM), model learns to ignore latent representation, i.e. $\mathrm{KL}(q(\mathbf{z}) \,||\, p(\mathbf{z})) \approx 0$.
- Want to use powerful $p_\theta(\mathbf{x} \mid \mathbf{z})$ to model the underlying data well, but also want to learn interesting representations $\mathbf{z}$.

VAE Issues: Posterior Collapse (Bowman al. 2016)

(1) Posterior collapse

- If generative model $p_\theta(\mathbf{x} \mid \mathbf{z})$ is too flexible (e.g. PixelCNN, LSTM), model learns to ignore latent representation, i.e. $\mathrm{KL}(q(\mathbf{z}) \,||\, p(\mathbf{z})) \approx 0$.
- Want to use powerful $p_\theta(\mathbf{x} \mid \mathbf{z})$ to model the underlying data well, but also want to learn interesting representations $\mathbf{z}$.

VAE Issues: Posterior Collapse (Bowman al. 2016)

(1) Posterior collapse

- If generative model $p_\theta(\mathbf{x} \,|\, \mathbf{z})$ is too flexible (e.g. PixelCNN, LSTM), model learns to ignore latent representation, i.e. $\mathrm{KL}(q(\mathbf{z}) \,||\, p(\mathbf{z})) \approx 0$.
- Want to use powerful $p_\theta(\mathbf{x} \,|\, \mathbf{z})$ to model the underlying data well, but also want to learn interesting representations $\mathbf{z}$.

Example: Text Modeling on Yahoo corpus (Yang et al. 2017)

Inference Network: LSTM + MLP

Generative Model: LSTM, $\mathbf{z}$ fed at each time step

Model	KL	PPL
Language Model	–	61.6
VAE	0.01	≤ 62.5
VAE + Word-Drop 25%	1.44	≤ 65.6
VAE + Word-Drop 50%	5.29	≤ 75.2
ConvNetVAE (Yang et al. 2017)	10.0	≤ 63.9

Example: Text Modeling on Yahoo corpus (Yang et al. 2017)

Inference Network: LSTM + MLP

Generative Model: LSTM, $\mathbf{z}$ fed at each time step

Model	KL	PPL
Language Model	–	61.6
VAE	0.01	≤ 62.5
VAE + Word-Drop 25%	1.44	≤ 65.6
VAE + Word-Drop 50%	5.29	≤ 75.2
ConvNetVAE (Yang et al. 2017)	10.0	≤ 63.9

VAE Issues: Inference Gap (Cremer et al. 2018)

(2) Inference Gap

Ideally, $q_{\mathsf{enc}_\phi(\mathbf{x})}(\mathbf{z}) \approx p_\theta(\mathbf{z} \,|\, \mathbf{x})$

$$\underbrace{\mathrm{KL}(q_{\mathsf{enc}_\phi(\mathbf{x})}(\mathbf{z}) \,||\, p_\theta(\mathbf{z} \,|\, \mathbf{x}))}_{\text{Inference gap}} = \underbrace{\mathrm{KL}(q_{\lambda^\star}(\mathbf{z}) \,||\, p_\theta(\mathbf{z} \,|\, \mathbf{x}))}_{\text{Approximation gap}} +$$

$$\underbrace{\mathrm{KL}(q_{\mathsf{enc}_\phi(\mathbf{x})}(\mathbf{z}) \,||\, p_\theta(\mathbf{z} \,|\, \mathbf{x})) - \mathrm{KL}(q_{\lambda^\star}(\mathbf{z}) \,||\, p_\theta(\mathbf{z} \,|\, \mathbf{x}))}_{\text{Amortization gap}}$$

- **Approximation gap**: Gap between true posterior and the best possible variational posterior $\lambda^\star$ cwithin $\mathcal{Q}$
- **Amortization gap**: Gap between the inference network posterior and best possible posterior

VAE Issues: Inference Gap (Cremer et al. 2018)

(2) Inference Gap

Ideally, $q_{\mathsf{enc}_\phi(\mathbf{x})}(\mathbf{z}) \approx p_\theta(\mathbf{z} \,|\, \mathbf{x})$

$$\underbrace{\mathrm{KL}(q_{\mathsf{enc}_\phi(\mathbf{x})}(\mathbf{z}) \,||\, p_\theta(\mathbf{z} \,|\, \mathbf{x}))}_{\text{Inference gap}} = \underbrace{\mathrm{KL}(q_{\lambda^\star}(\mathbf{z}) \,||\, p_\theta(\mathbf{z} \,|\, \mathbf{x}))}_{\text{Approximation gap}} +$$

$$\underbrace{\mathrm{KL}(q_{\mathsf{enc}_\phi(\mathbf{x})}(\mathbf{z}) \,||\, p_\theta(\mathbf{z} \,|\, \mathbf{x})) - \mathrm{KL}(q_{\lambda^\star}(\mathbf{z}) \,||\, p_\theta(\mathbf{z} \,|\, \mathbf{x}))}_{\text{Amortization gap}}$$

- **Approximation gap**: Gap between true posterior and the best possible variational posterior $\lambda^\star$ cwithin $\mathcal{Q}$
- **Amortization gap**: Gap between the inference network posterior and best possible posterior

VAE Issues (Cremer et al. 2018)

- These gaps affect the learned generative model.
- **Approximation gap**: use more flexible variational families, e.g. Normalizing/IA Flows (Rezende et al. 2015, Kingma et al. 2016)
  $\implies$ Has not been show to fix posterior collapse on text.
- **Amortization gap**: better optimize λ for each data point, e.g. with iterative inference (Hjelm et al. 2016, Krishnan et al. 2018)
  $\implies$ Focus of this work.
- Does reducing the amortization gap allow us to employ powerful likelihood models while avoiding posterior collapse?

VAE Issues (Cremer et al. 2018)

- These gaps affect the learned generative model.
- **Approximation gap**: use more flexible variational families, e.g. Normalizing/IA Flows (Rezende et al. 2015, Kingma et al. 2016)
  $\implies$ Has not been show to fix posterior collapse on text.
- **Amortization gap**: better optimize λ for each data point, e.g. with iterative inference (Hjelm et al. 2016, Krishnan et al. 2018)
  $\implies$ Focus of this work.
- Does reducing the amortization gap allow us to employ powerful likelihood models while avoiding posterior collapse?

VAE Issues (Cremer et al. 2018)

- These gaps affect the learned generative model.
- **Approximation gap**: use more flexible variational families, e.g. Normalizing/IA Flows (Rezende et al. 2015, Kingma et al. 2016)
  $\implies$ Has not been show to fix posterior collapse on text.
- **Amortization gap**: better optimize λ for each data point, e.g. with iterative inference (Hjelm et al. 2016, Krishnan et al. 2018)
  $\implies$ Focus of this work.
- Does reducing the amortization gap allow us to employ powerful likelihood models while avoiding posterior collapse?

VAE Issues (Cremer et al. 2018)

- These gaps affect the learned generative model.
- **Approximation gap**: use more flexible variational families, e.g. Normalizing/IA Flows (Rezende et al. 2015, Kingma et al. 2016)
  $\implies$ Has not been show to fix posterior collapse on text.
- **Amortization gap**: better optimize λ for each data point, e.g. with iterative inference (Hjelm et al. 2016, Krishnan et al. 2018)
  $\implies$ Focus of this work.
- Does reducing the amortization gap allow us to employ powerful likelihood models while avoiding posterior collapse?

Stochastic Variational Inference (SVI) (Hoffman et al. 2013)

- Amortization gap is mostly specific to VAE
- Stochastic Variational Inference (SVI):
  1. Randomly initialize $\lambda_0^{(i)}$ for each data point
  2. Perform iterative inference, e.g. for $k = 1, \dots, K$

$$\lambda_k^{(i)} \leftarrow \lambda_{k-1}^{(i)} - \alpha \nabla_\lambda \mathcal{L}(\lambda_k^{(i)}, \theta, \mathbf{x}^{(i)})$$

     where $\mathcal{L}(\lambda, \theta, \mathbf{x}) = \mathbb{E}_{q_\lambda(\mathbf{z})}[-\log p_\theta(\mathbf{x} \mid \mathbf{z})] + \mathrm{KL}(q_\lambda(\mathbf{z}) \,||\, p(\mathbf{z})]$
  3. Update θ based on final $\lambda_K^{(i)}$, i.e.

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\lambda_K^{(i)}, \theta, \mathbf{x}^{(i)})$$

(Can reduce amortization gap by increasing K)

Stochastic Variational Inference (SVI) (Hoffman et al. 2013)

- Amortization gap is mostly specific to VAE
- Stochastic Variational Inference (SVI):
  1. Randomly initialize $\lambda_0^{(i)}$ for each data point
  2. Perform iterative inference, e.g. for $k = 1, \dots, K$

$$\lambda_k^{(i)} \leftarrow \lambda_{k-1}^{(i)} - \alpha \nabla_\lambda \mathcal{L}(\lambda_k^{(i)}, \theta, \mathbf{x}^{(i)})$$

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

  3. Update θ based on final $\lambda_K^{(i)}$, i.e.

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\lambda_K^{(i)}, \theta, \mathbf{x}^{(i)})$$

(Can reduce amortization gap by increasing K)

Stochastic Variational Inference (SVI) (Hoffman et al. 2013)

- Amortization gap is mostly specific to VAE
- Stochastic Variational Inference (SVI):
  1. Randomly initialize $\lambda_0^{(i)}$ for each data point
  2. Perform iterative inference, e.g. for $k = 1, \dots, K$

$$\lambda_k^{(i)} \leftarrow \lambda_{k-1}^{(i)} - \alpha \nabla_\lambda \mathcal{L}(\lambda_k^{(i)}, \theta, \mathbf{x}^{(i)})$$

     where $\mathcal{L}(\lambda, \theta, \mathbf{x}) = \mathbb{E}_{q_\lambda(\mathbf{z})}[-\log p_\theta(\mathbf{x} \,|\, \mathbf{z})] + \mathrm{KL}(q_\lambda(\mathbf{z}) \,||\, p(\mathbf{z}))$
  3. Update θ based on final $\lambda_K^{(i)}$, i.e.

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\lambda_K^{(i)}, \theta, \mathbf{x}^{(i)})$$

(Can reduce amortization gap by increasing K)

Stochastic Variational Inference (SVI) (Hoffman et al. 2013)

- Amortization gap is mostly specific to VAE
- Stochastic Variational Inference (SVI):
  1. Randomly initialize $\lambda_0^{(i)}$ for each data point
  2. Perform iterative inference, e.g. for $k = 1, \ldots, K$

$$\lambda_k^{(i)} \leftarrow \lambda_{k-1}^{(i)} - \alpha \nabla_\lambda \mathcal{L}(\lambda_k^{(i)}, \theta, \mathbf{x}^{(i)})$$

     where $\mathcal{L}(\lambda, \theta, \mathbf{x}) = \mathbb{E}_{q_\lambda(\mathbf{z})}[-\log p_\theta(\mathbf{x} \,|\, \mathbf{z})] + \mathrm{KL}(q_\lambda(\mathbf{z}) \,||\, p(\mathbf{z}))$
  3. Update θ based on final $\lambda_K^{(i)}$, i.e.

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\lambda_K^{(i)}, \theta, \mathbf{x}^{(i)})$$

(Can reduce amortization gap by increasing K)

Example: Text Modeling on Yahoo corpus (Yang et al. 2017)

Inference Network: LSTM + MLP

Generative Model: LSTM, $\mathbf{z}$ fed at each time step

Model	KL	PPL
Language Model	–	61.6
VAE	0.01	≤ 62.5
SVI ($K = 20$)	0.41	≤ 62.9
SVI ($K = 40$)	1.01	≤ 62.2

Comparing the Amortized/Stochastic Variational Inference

	AVI	SVI
Approximation Gap	Yes	Yes
Amortization Gap	Yes	Minimal
Training/Inference	Fast	Slow
End-to-End Training	Yes	No

SVI: Trade-off between amortization gap vs speed

This Work: Semi-Amortized Variational Autoencoders

- Reduce amortization gap in VAEs by combining AVI/SVI
- Use inference network to initialize variational parameters, run SVI to refine them
- Maintain end-to-end training of VAEs by backpropagating through SVI to train the inference network/generative model

This Work: Semi-Amortized Variational Autoencoders

- Reduce amortization gap in VAEs by combining AVI/SVI
- Use inference network to initialize variational parameters, run SVI to refine them
- Maintain end-to-end training of VAEs by backpropagating through SVI to train the inference network/generative model

This Work: Semi-Amortized Variational Autoencoders

- Reduce amortization gap in VAEs by combining AVI/SVI
- Use inference network to initialize variational parameters, run SVI to refine them
- Maintain end-to-end training of VAEs by backpropagating through SVI to train the inference network/generative model

Semi-Amortized Variational Autoencoders (SA-VAE)

Forward step

1. $\lambda_0 = \text{enc}_\phi(\mathbf{x})$
2. For $k = 1, \dots, K$

$$\lambda_k \leftarrow \lambda_{k-1} - \alpha \nabla_\lambda \mathcal{L}(\lambda_k, \theta, \mathbf{x})$$

where $\mathcal{L}(\lambda, \theta, \mathbf{x}) = \mathbb{E}_{q_\lambda(\mathbf{z})}[-\log p_\theta(\mathbf{x} \,|\, \mathbf{z})] + \mathrm{KL}(q_\lambda(\mathbf{z}) \,||\, p(\mathbf{z}))$

3. Final loss given by

$$L_K = \mathcal{L}(\lambda_K, \theta, \mathbf{x})$$

Semi-Amortized Variational Autoencoders (SA-VAE)

Forward step

1. $\lambda_0 = \text{enc}_\phi(\mathbf{x})$
2. For $k = 1, \dots, K$

$$\lambda_k \leftarrow \lambda_{k-1} - \alpha \nabla_\lambda \mathcal{L}(\lambda_k, \theta, \mathbf{x})$$

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

3. Final loss given by

$$L_K = \mathcal{L}(\lambda_K, \theta, \mathbf{x})$$

Semi-Amortized Variational Autoencoders (SA-VAE)

Forward step

1. $\lambda_0 = \text{enc}_\phi(\mathbf{x})$
2. For $k = 1, \dots, K$

$$\lambda_k \leftarrow \lambda_{k-1} - \alpha \nabla_\lambda \mathcal{L}(\lambda_k, \theta, \mathbf{x})$$

   where $\mathcal{L}(\lambda, \theta, \mathbf{x}) = \mathbb{E}_{q_\lambda(\mathbf{z})}[-\log p_\theta(\mathbf{x} \,|\, \mathbf{z})] + \mathrm{KL}(q_\lambda(\mathbf{z}) \,||\, p(\mathbf{z}))$

3. Final loss given by

$$L_K = \mathcal{L}(\lambda_K, \theta, \mathbf{x})$$

Semi-Amortized Variational Autoencoders (SA-VAE)

Backward step

- Need to calculate derivative of L_K with respect to θ, ϕ
- But $\lambda_1, \dots \lambda_K$ are all functions of θ, ϕ

$$\begin{aligned}
\lambda_K &= \lambda_{K-1} - \alpha \nabla_\lambda \mathcal{L}(\lambda_{K-1}, \theta, x) \\
&= \lambda_{K-2} - \alpha \nabla_\lambda \mathcal{L}(\lambda_{K-2}, \theta, x) \\
&\quad - \alpha \nabla_\lambda \mathcal{L}(\lambda_{K-2} - \alpha \nabla_\lambda \mathcal{L}(\lambda_{K-2}, \theta, x), \theta, x) \\
&= \lambda_{K-3} - \dots
\end{aligned}$$

- Calculating the total derivative requires "unrolling optimization" and backpropagating through gradient descent (Domke 2012, Maclaurin et al. 2015, Belanger et al. 2017).

Semi-Amortized Variational Autoencoders (SA-VAE)

Backward step

- Need to calculate derivative of L_K with respect to θ, ϕ
- But $\lambda_1, \dots \lambda_K$ are all functions of θ, ϕ

$$
\begin{aligned}
\lambda_K &= \lambda_{K-1} - \alpha \nabla_\lambda \mathcal{L}(\lambda_{K-1}, \theta, x) \\
&= \lambda_{K-2} - \alpha \nabla_\lambda \mathcal{L}(\lambda_{K-2}, \theta, x) \\
&\quad - \alpha \nabla_\lambda \mathcal{L}(\lambda_{K-2} - \alpha \nabla_\lambda \mathcal{L}(\lambda_{K-2}, \theta, x), \theta, x) \\
&= \lambda_{K-3} - \dots
\end{aligned}
$$

- Calculating the total derivative requires “unrolling optimization” and backpropagating through gradient descent (Domke 2012, Maclaurin et al. 2015, Belanger et al. 2017).

Backpropagating through SVI

Simple example: consider just one step of SVI

1. $\lambda_0 = \mathsf{enc}_\phi(\mathbf{x})$
2. $\lambda_1 = \lambda_0 - \alpha \nabla_\lambda \mathcal{L}(\lambda_0, \theta, \mathbf{x})$
3. $L = \mathcal{L}(\lambda_1, \theta, \mathbf{x})$

Backpropagating through SVI

Backward step

1. Calculate $\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}$
2. Chain rule:

$$\frac{\mathrm{d}L}{\mathrm{d}\lambda_0} = \frac{\mathrm{d}\lambda_1}{\mathrm{d}\lambda_0}\frac{\mathrm{d}L}{\mathrm{d}\lambda_1} = \frac{\mathrm{d}}{\mathrm{d}\lambda_0}\Big(\lambda_0 - \alpha\nabla_\lambda \mathcal{L}(\lambda_0, \theta, \mathbf{x})\Big)\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}$$
$$= \Big(\mathbf{I} - \alpha \underbrace{\nabla^2_\lambda \mathcal{L}(\lambda_0, \theta, \mathbf{x})}_{\text{Hessian matrix}}\Big)\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}$$
$$= \frac{\mathrm{d}L}{\mathrm{d}\lambda_1} - \alpha \underbrace{\nabla^2_\lambda \mathcal{L}(\lambda_0, \theta, \mathbf{x})\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}}_{\text{Hessian-vector product}}$$

3. Backprop $\frac{\mathrm{d}L}{\mathrm{d}\lambda_0}$ to obtain $\frac{\mathrm{d}L}{\mathrm{d}\phi} = \frac{\mathrm{d}\lambda_0}{\mathrm{d}\phi}\frac{\mathrm{d}L}{\mathrm{d}\lambda_0}$ (Similar rules for $\frac{\mathrm{d}L}{\mathrm{d}\theta}$)

Backpropagating through SVI

Backward step

1. Calculate $\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}$
2. Chain rule:

$$\frac{\mathrm{d}L}{\mathrm{d}\lambda_0} = \frac{\mathrm{d}\lambda_1}{\mathrm{d}\lambda_0}\frac{\mathrm{d}L}{\mathrm{d}\lambda_1} = \frac{\mathrm{d}}{\mathrm{d}\lambda_0}\Big(\lambda_0 - \alpha\nabla_\lambda\mathcal{L}(\lambda_0,\theta,\mathbf{x})\Big)\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}$$

$$= \Big(\mathbf{I} - \alpha\underbrace{\nabla^2_\lambda\mathcal{L}(\lambda_0,\theta,\mathbf{x})}_{\text{Hessian matrix}}\Big)\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}$$

$$= \frac{\mathrm{d}L}{\mathrm{d}\lambda_1} - \alpha\underbrace{\nabla^2_\lambda\mathcal{L}(\lambda_0,\theta,\mathbf{x})\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}}_{\text{Hessian-vector product}}$$

3. Backprop $\frac{\mathrm{d}L}{\mathrm{d}\lambda_0}$ to obtain $\frac{\mathrm{d}L}{\mathrm{d}\phi} = \frac{\mathrm{d}\lambda_0}{\mathrm{d}\phi}\frac{\mathrm{d}L}{\mathrm{d}\lambda_0}$ (Similar rules for $\frac{\mathrm{d}L}{\mathrm{d}\theta}$)

Backpropagating through SVI

Backward step

1. Calculate $\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}$
2. Chain rule:

$$\frac{\mathrm{d}L}{\mathrm{d}\lambda_0} = \frac{\mathrm{d}\lambda_1}{\mathrm{d}\lambda_0}\frac{\mathrm{d}L}{\mathrm{d}\lambda_1} = \frac{\mathrm{d}}{\mathrm{d}\lambda_0}\Big(\lambda_0 - \alpha\nabla_\lambda \mathcal{L}(\lambda_0, \theta, \mathbf{x})\Big)\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}$$

$$= \Big(\mathbf{I} - \alpha \underbrace{\nabla_\lambda^2 \mathcal{L}(\lambda_0, \theta, \mathbf{x})}_{\text{Hessian matrix}}\Big)\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}$$

$$= \frac{\mathrm{d}L}{\mathrm{d}\lambda_1} - \alpha \underbrace{\nabla_\lambda^2 \mathcal{L}(\lambda_0, \theta, \mathbf{x})\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}}_{\text{Hessian-vector product}}$$

3. Backprop $\frac{\mathrm{d}L}{\mathrm{d}\lambda_0}$ to obtain $\frac{\mathrm{d}L}{\mathrm{d}\phi} = \frac{\mathrm{d}\lambda_0}{\mathrm{d}\phi}\frac{\mathrm{d}L}{\mathrm{d}\lambda_0}$ (Similar rules for $\frac{\mathrm{d}L}{\mathrm{d}\theta}$)

Backpropagating through SVI

Backward step

1. Calculate $\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}$
2. Chain rule:

$$
\begin{aligned}
\frac{\mathrm{d}L}{\mathrm{d}\lambda_0} &= \frac{\mathrm{d}\lambda_1}{\mathrm{d}\lambda_0}\frac{\mathrm{d}L}{\mathrm{d}\lambda_1} = \frac{\mathrm{d}}{\mathrm{d}\lambda_0}\Big(\lambda_0 - \alpha\nabla_\lambda \mathcal{L}(\lambda_0, \theta, \mathbf{x})\Big)\frac{\mathrm{d}L}{\mathrm{d}\lambda_1} \\
&= \Big(\mathbf{I} - \alpha\,\underbrace{\nabla^2_\lambda \mathcal{L}(\lambda_0, \theta, \mathbf{x})}_{\text{Hessian matrix}}\Big)\frac{\mathrm{d}L}{\mathrm{d}\lambda_1} \\
&= \frac{\mathrm{d}L}{\mathrm{d}\lambda_1} - \alpha\,\underbrace{\nabla^2_\lambda \mathcal{L}(\lambda_0, \theta, \mathbf{x})\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}}_{\text{Hessian-vector product}}
\end{aligned}
$$

3. Backprop $\frac{\mathrm{d}L}{\mathrm{d}\lambda_0}$ to obtain $\frac{\mathrm{d}L}{\mathrm{d}\phi} = \frac{\mathrm{d}\lambda_0}{\mathrm{d}\phi}\frac{\mathrm{d}L}{\mathrm{d}\lambda_0}$ (Similar rules for $\frac{\mathrm{d}L}{\mathrm{d}\theta}$)

Backpropagating through SVI

Backward step

1. Calculate $\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}$
2. Chain rule:

$$\begin{aligned}\frac{\mathrm{d}L}{\mathrm{d}\lambda_0} &= \frac{\mathrm{d}\lambda_1}{\mathrm{d}\lambda_0}\frac{\mathrm{d}L}{\mathrm{d}\lambda_1} = \frac{\mathrm{d}}{\mathrm{d}\lambda_0}\Big(\lambda_0 - \alpha\nabla_\lambda \mathcal{L}(\lambda_0, \theta, \mathbf{x})\Big)\frac{\mathrm{d}L}{\mathrm{d}\lambda_1} \\ &= \Big(\mathbf{I} - \alpha\underbrace{\nabla^2_\lambda \mathcal{L}(\lambda_0, \theta, \mathbf{x})}_{\text{Hessian matrix}}\Big)\frac{\mathrm{d}L}{\mathrm{d}\lambda_1} \\ &= \frac{\mathrm{d}L}{\mathrm{d}\lambda_1} - \alpha\underbrace{\nabla^2_\lambda \mathcal{L}(\lambda_0, \theta, \mathbf{x})\frac{\mathrm{d}L}{\mathrm{d}\lambda_1}}_{\text{Hessian-vector product}}\end{aligned}$$

3. Backprop $\frac{\mathrm{d}L}{\mathrm{d}\lambda_0}$ to obtain $\frac{\mathrm{d}L}{\mathrm{d}\phi} = \frac{\mathrm{d}\lambda_0}{\mathrm{d}\phi}\frac{\mathrm{d}L}{\mathrm{d}\lambda_0}$ (Similar rules for $\frac{\mathrm{d}L}{\mathrm{d}\theta}$)

Backpropagating through SVI

In practice:

- Estimate Hessian-vector products with finite differences (LeCun et al. 1993), which was more memory efficient.
- Clip gradients at various points (see paper).

Summary

	AVI	SVI	SA-VAE
Approximation Gap	Yes	Yes	Yes
Amortization Gap	Yes	Minimal	Minimal
Training/Inference	Fast	Slow	Medium
End-to-End Training	Yes	No	Yes

Experiments: Synthetic data

Generate sequential data from a randomly initialized LSTM oracle

1. $z_1, z_2 \sim \mathcal{N}(0, 1)$
2. $h_t = \mathrm{LSTM}([x_t, z_1, z_2], h_{t-1})$
3. $p(x_{t+1} \,|\, x_{\leq t}, \mathbf{z}) \propto \exp(\mathbf{W} h_t)$

Inference network

- $q(z_1), q(z_2)$ are Gaussians with learned means $\mu_1, \mu_2 = \mathrm{enc}_\phi(\mathbf{x})$
- $\mathrm{enc}_\phi(\cdot)$: LSTM with MLP on final hidden state

Experiments: Synthetic data

Generate sequential data from a randomly initialized LSTM oracle

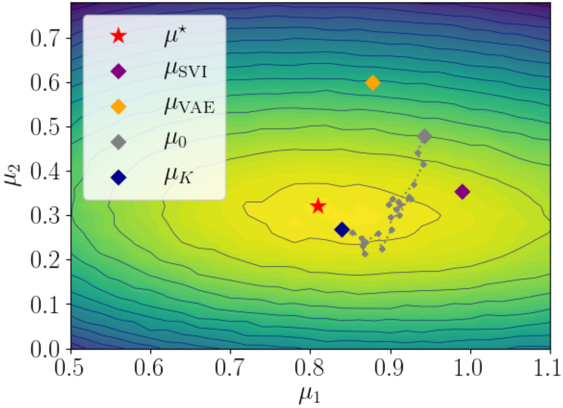
1. $z_1, z_2 \sim \mathcal{N}(0, 1)$
2. $h_t = \mathrm{LSTM}([x_t, z_1, z_2], h_{t-1})$
3. $p(x_{t+1} \,|\, x_{\leq t}, \mathbf{z}) \propto \exp(\mathbf{W} h_t)$

Inference network

- $q(z_1), q(z_2)$ are Gaussians with learned means $\mu_1, \mu_2 = \mathsf{enc}_\phi(\mathbf{x})$
- $\mathsf{enc}_\phi(\cdot)$: LSTM with MLP on final hidden state

Experiments: Synthetic data

Oracle generative model (randomly-initialized LSTM)



(ELBO landscape for a random test point)

Results: Synthetic Data

Model	Oracle Gen	Learned Gen
VAE	≤ 21.77	≤ 27.06
SVI (K=20)	≤ 22.33	≤ 25.82
SA-VAE (K=20)	≤ 20.13	≤ 25.21
True NLL (Est)	19.63	–

Results: Text

Generative model:

1. $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2. $h_t = \mathrm{LSTM}([x_t, \mathbf{z}], h_{t-1})$
3. $x_{t+1} \sim p(x_{t+1} \,|\, x_{\leq t}, \mathbf{x}) \propto \exp(\mathbf{W} h_t)$

Inference network:

- $q(\mathbf{z})$ diagonal Gaussian with parameters $\boldsymbol{\mu}, \boldsymbol{\sigma}^{\mathbf{2}}$
- $\boldsymbol{\mu}, \boldsymbol{\sigma}^{\mathbf{2}} = \mathsf{enc}_\phi(\mathbf{x})$
- $\mathsf{enc}_\phi(\cdot)$: LSTM followed by MLP

Results: Text

Two other baselines that combine AVI/SVI (but not end-to-end):

- VAE+SVI 1 (Krishnan et a al. 2018):
  1. Update generative model based on λ_K
  2. Update inference network based on λ_0
- VAE+SVI 2 (Hjelm et al. 2016):
  1. Update generative model based on λ_K
  2. Update inference network to minimize $\mathrm{KL}(q_{\lambda_0}(\mathbf{z}) \,\|\, q_{\lambda_K}(\mathbf{z}))$, treating λ_K as a fixed constant.

(Forward pass is the same for both models)

Results: Text

Two other baselines that combine AVI/SVI (but not end-to-end):

- VAE+SVI 1 (Krishnan et a al. 2018):
  1. Update generative model based on λ_K
  2. Update inference network based on λ_0
- VAE+SVI 2 (Hjelm et al. 2016):
  1. Update generative model based on λ_K
  2. Update inference network to minimize $\mathrm{KL}(q_{\lambda_0}(\mathbf{z}) \,\|\, q_{\lambda_K}(\mathbf{z}))$, treating λ_K as a fixed constant.

(Forward pass is the same for both models)

Results: Text

Two other baselines that combine AVI/SVI (but not end-to-end):

- VAE+SVI 1 (Krishnan et a al. 2018):
  1. Update generative model based on λ_K
  2. Update inference network based on λ_0
- VAE+SVI 2 (Hjelm et al. 2016):
  1. Update generative model based on λ_K
  2. Update inference network to minimize $\mathrm{KL}(q_{\lambda_0}(\mathbf{z}) \,\|\, q_{\lambda_K}(\mathbf{z}))$, treating λ_K as a fixed constant.

(Forward pass is the same for both models)

Results: Text (Yahoo corpus from Yang et al. 2017)

Model	KL	PPL
Language Model	–	61.6
VAE	0.01	≤ 62.5
VAE + Word-Drop 25%	1.44	≤ 65.6
VAE + Word-Drop 50%	5.29	≤ 75.2
ConvNetVAE (Yang et al. 2017)	10.0	≤ 63.9
SVI ($K = 20$)	0.41	≤ 62.9
SVI ($K = 40$)	1.01	≤ 62.2
VAE + SVI 1 ($K = 20$)	7.80	≤ 62.7
VAE + SVI 2 ($K = 20$)	7.81	≤ 62.3
SA-VAE ($K = 20$)	7.19	≤ 60.4

Results: Text (Yahoo corpus from Yang et al. 2017)

MODEL	KL	PPL
LANGUAGE MODEL	–	61.6
VAE	0.01	≤ 62.5
VAE + WORD-DROP 25%	1.44	≤ 65.6
VAE + WORD-DROP 50%	5.29	≤ 75.2
CONVNETVAE (YANG ET AL. 2017)	10.0	≤ 63.9
SVI ($K = 20$)	0.41	≤ 62.9
SVI ($K = 40$)	1.01	≤ 62.2
VAE + SVI 1 ($K = 20$)	7.80	≤ 62.7
VAE + SVI 2 ($K = 20$)	7.81	≤ 62.3
SA-VAE ($K = 20$)	7.19	≤ 60.4

Results: Text (Yahoo corpus from Yang et al. 2017)

Model	KL	PPL
Language Model	–	61.6
VAE	0.01	≤ 62.5
VAE + Word-Drop 25%	1.44	≤ 65.6
VAE + Word-Drop 50%	5.29	≤ 75.2
ConvNetVAE (Yang et al. 2017)	10.0	≤ 63.9
SVI ($K = 20$)	0.41	≤ 62.9
SVI ($K = 40$)	1.01	≤ 62.2
VAE + SVI 1 ($K = 20$)	7.80	≤ 62.7
VAE + SVI 2 ($K = 20$)	7.81	≤ 62.3
SA-VAE ($K = 20$)	7.19	≤ 60.4

Application to Image Modeling (OMNIGLOT)

$q_\phi(\mathbf{z} \mid \mathbf{x})$: 3-layer ResNet (He et al. 2016)

$p_\theta(\mathbf{x} \mid \mathbf{z})$: 12-layer Gated PixelCNN (van den Oord et al. 2016)

Model	NLL (KL)
Gated PixelCNN	90.59
VAE	$\leq$ 90.43 (0.98)
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 1($K = 20$)	$\leq$ 90.19 (2.40)
VAE + SVI 2 ($K = 20$)	$\leq$ 90.21 (2.83)
SA-VAE ($K = 20$)	$\leq$ 90.05 (2.78)

(Amortization gap exists even with powerful inference networks)

Application to Image Modeling (OMNIGLOT)

$q_\phi(\mathbf{z} \mid \mathbf{x})$: 3-layer ResNet (He et al. 2016)

$p_\theta(\mathbf{x} \mid \mathbf{z})$: 12-layer Gated PixelCNN (van den Oord et al. 2016)

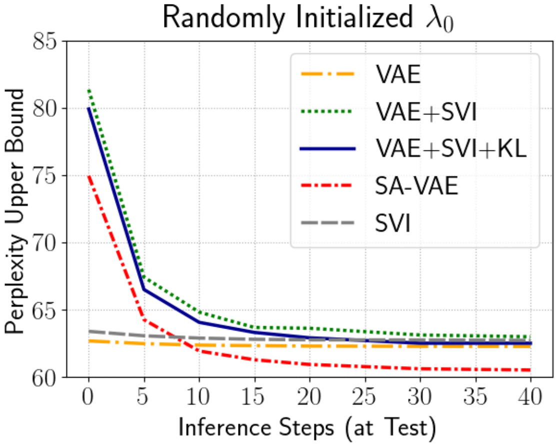
Model	NLL (KL)
Gated PixelCNN	90.59
VAE	$\leq$ 90.43 (0.98)
SVI ($K = 20$)	$\leq$ 90.51 (0.06)
SVI ($K = 40$)	$\leq$ 90.44 (0.27)
SVI ($K = 80$)	$\leq$ 90.27 (1.65)
VAE + SVI 1($K = 20$)	$\leq$ 90.19 (2.40)
VAE + SVI 2 ($K = 20$)	$\leq$ 90.21 (2.83)
SA-VAE ($K = 20$)	$\leq$ 90.05 (2.78)

(Amortization gap exists even with powerful inference networks)

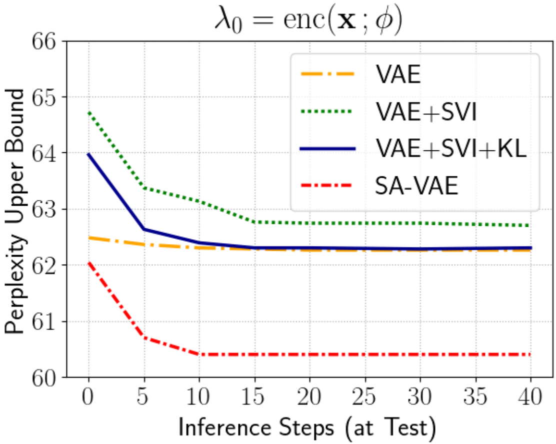
Limitations

- Requires $O(K)$ backpropagation steps of the generative model for each training setup: possible to reduce K via
  - Learning to learn approaches
  - Dynamic scheduling
  - Importance sampling
- Still needs optimization hacks
  - Gradient clipping during iterative refinement

Train vs Test Analysis



Train vs Test Analysis



Lessons Learned

- Reducing amortization gap helps learn generative models of text that give good likelihoods and maintains interesting latent representations.
- But certainly not the full story... still very much an open issue.
- So what are the latent variables capturing?

Lessons Learned

- Reducing amortization gap helps learn generative models of text that give good likelihoods and maintains interesting latent representations.
- But certainly not the full story... still very much an open issue.
- So what are the latent variables capturing?

Lessons Learned

- Reducing amortization gap helps learn generative models of text that give good likelihoods and maintains interesting latent representations.
- But certainly not the full story... still very much an open issue.
- So what are the latent variables capturing?

Saliency Analysis

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

Generations

Test sentence in blue, two generations from $q(\mathbf{z} \mid \mathbf{x})$ in red

<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>

Generations

Test sentence in blue, two generations from $q(\mathbf{z} \mid \mathbf{x})$ in red

<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>

Generations

New sentence in blue, two generations from $q(\mathbf{z} \mid \mathbf{x})$ in red

<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>

Generations

New sentence in blue, two generations from $q(\mathbf{z} \mid \mathbf{x})$ in red

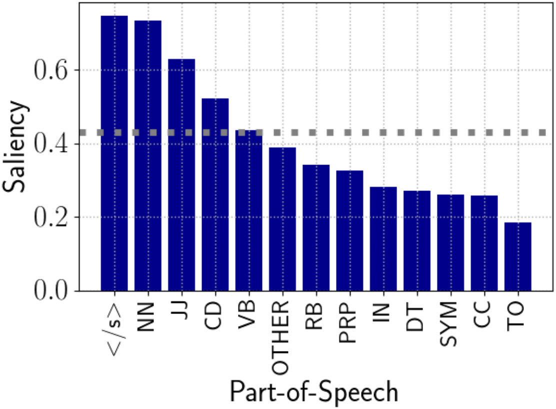
<s> which country is the best at soccer ? brazil or germany . </s>

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>

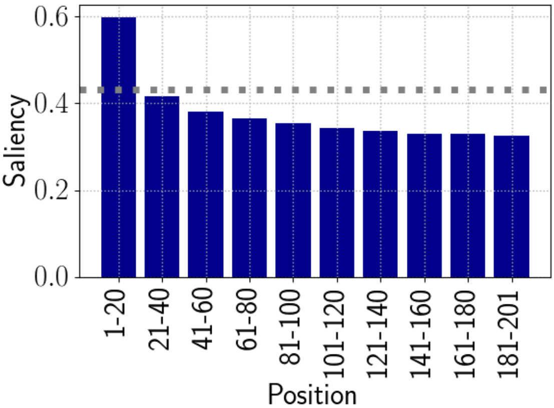
Saliency Analysis

Saliency analysis by Part-of-Speech Tag



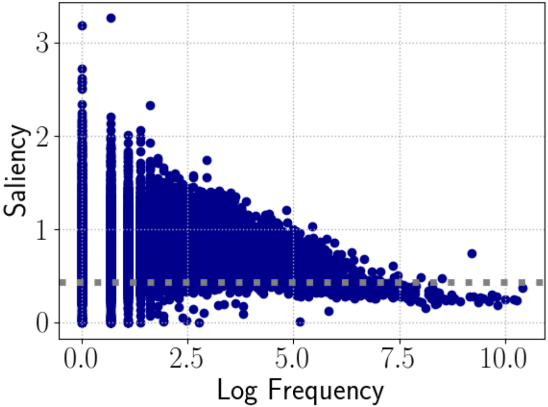
Saliency Analysis

Saliency analysis by Position



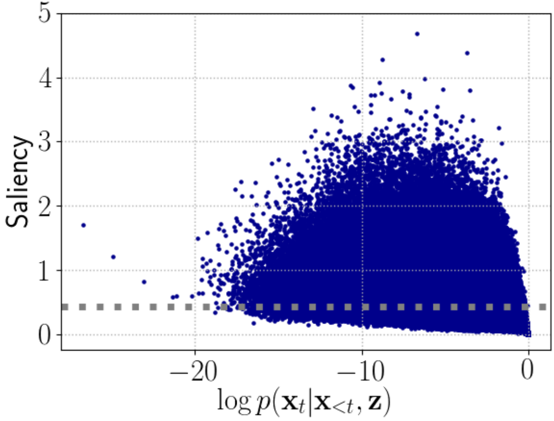
Saliency Analysis

Saliency analysis by Frequency



Saliency Analysis

Saliency analysis by PPL



Conclusion

- Reducing amortization gap helps learn generative models that better utilize the latent space.
- Can be combined with methods that reduce the approximation gap.