A Surprisingly Effective Fix for Deep Latent Variable Modeling of Text

Bohan Li*[1], Junxian He*[1], Graham Neubig[1], Taylor Berg-Kirkpatrick[2], Yiming Yang[1]

Core idea:

- **Pretraining encoder using AE objective**
- **Initializing VAE with pertained AE and training with FB objective**

Results:

		Yahoo			
LSTM-LM	60.75	-	-	-	-
VAE	61.52	329.10	0	0.00	329.10
+ anneal	61.21	328.80	0	0.00	328.80
+ cyclic	66.93	333.80	4	2.83	336.63
+ aggressive	59.77	322.70	15	5.70	328.40
+ FBP ($\lambda = 9$)	62.59	322.91	6	9.08	331.99
+ FBP ($\lambda = 7$)	62.76	324.66	5	7.03	331.69
+ FBP ($\lambda = 5$)	62.78	326.26	3	5.07	331.32
+ FBP ($\lambda = 3$)	62.88	328.13	2	3.06	331.19
Ours ($\lambda = 6$)	59.23	317.39	32	12.09	329.48
Ours ($\lambda = 8$)	**59.51**	**315.31**	**32**	15.02	330.33
Ours ($\lambda = 9$)	59.60	315.09	32	15.49	330.58

Implicit Deep Latent Variable Models for Text Generation (EMNLP 2019)

Le Fang[†], Chunyuan Li[§], Jianfeng Gao[§], Wen Dong[†], Changyou Chen[†]

Core ideas:

- Attributing posterior collapse to the restrictive Gaussian assumption, and advocate more flexible sample-based posterior representation.
- Proposed iVAE:
    - Instead of assuming posterior as Gaussian, use sampling mechanism for $q_\phi(z|x)$. $z_{x,i} = G_\phi(x, \varepsilon_i)$, $\varepsilon_i \sim q(\varepsilon)$
    - How to evaluate KL on this? Use dual form of $\mathrm{KL}(q_\phi(z|x) \parallel p(z))$:

$$\begin{aligned}&\mathrm{KL}\left(q_{\phi}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z})\right) \\ =&\max_{\nu}\mathbb{E}_{\boldsymbol{z}\sim q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}\nu_{\psi}(\boldsymbol{x},\boldsymbol{z}) - \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\exp(\nu_{\psi}(\boldsymbol{x},\boldsymbol{z})),\end{aligned} \tag{7}$$

    - New iVAE objective:

$$\begin{aligned}\mathcal{L}_{\text{iVAE}} =& \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{z}\sim q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) \\ &- \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{z}\sim q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}\nu_{\psi}(\boldsymbol{x},\boldsymbol{z}) \\ &+ \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\exp(\nu_{\psi}(\boldsymbol{x},\boldsymbol{z})),\end{aligned} \tag{8}$$

    - Here, ν_ψ is a MLP taking in (x, z).

Implicit Deep Latent Variable Models for Text Generation (EMNLP 2019)

Le Fang[†], Chunyuan Li[§], Jianfeng Gao[§], Wen Dong[†], Changyou Chen[†]

Core ideas:

- Training Scheme:

  - Sample a mini-batch of $\boldsymbol{x}_i \sim \mathcal{D}$, $\epsilon_i \sim q(\epsilon)$, and generate $\boldsymbol{z}_{\boldsymbol{x}_i,\epsilon_i} = G(\boldsymbol{x}_i, \epsilon_i; \boldsymbol{\phi})$; Sample a mini-batch of $\boldsymbol{z}_i \sim p(\boldsymbol{z})$.

  - Update $\boldsymbol{\psi}$ in $\nu_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z})$ to maximize

$$\sum_i \nu_{\boldsymbol{\psi}}(\boldsymbol{x}_i, \boldsymbol{z}_{\boldsymbol{x}_i,\epsilon_i}) - \sum_i \exp(\nu_{\boldsymbol{\psi}}(\boldsymbol{x}_i, \boldsymbol{z}_i)) \quad (9)$$

  - Update parameters $\{\boldsymbol{\phi}, \boldsymbol{\theta}\}$ to maximize

$$\sum_i \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i | \boldsymbol{z}_{\boldsymbol{x}_i,\epsilon_i}) - \sum_i \nu_{\boldsymbol{\psi}}(\boldsymbol{x}_i, \boldsymbol{z}_{\boldsymbol{x}_i,\epsilon_i}) \quad (10)$$

Implicit Deep Latent Variable Models for Text Generation (EMNLP 2019)

Le Fang[†], Chunyuan Li[§], Jianfeng Gao[§], Wen Dong[†], Changyou Chen[†]

Core ideas:

- Proposed Mutual Information Regularized iVAE
    - Replace $-\mathrm{KL}(q_\phi(z|x) \parallel p(z))$ with $-\mathrm{KL}(q_\phi(z) \parallel p(z))$, where $q_\phi(z)=\int q(x)q_\phi(z|x)dx$, estimated by ancestral sampling in practice.
    - This objective also maximize the mutual information I(x,z), as they claimed.
    - The new objective:

$$\begin{aligned}&\mathrm{KL}\left(q_{\boldsymbol{\phi}}(\boldsymbol{z}) \parallel p(\boldsymbol{z})\right) \qquad (13)\\ =&\max_\nu \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\nu_{\boldsymbol{\psi}}(\boldsymbol{z}) - \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\exp(\nu_{\boldsymbol{\psi}}(\boldsymbol{z})).\end{aligned}$$

$$\begin{aligned}\mathcal{L}_{\mathrm{iVAE_{MI}}} =& \mathbb{E}_{\boldsymbol{x}\sim D}\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) \qquad (14)\\ &- \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\nu_{\boldsymbol{\psi}}(\boldsymbol{z}) + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\exp(\nu_{\boldsymbol{\psi}}(\boldsymbol{z})),\end{aligned}$$

    - The only difference with iVAE is that ν_ψ is a MLP taking in only z.

Implicit Deep Latent Variable Models for Text Generation (EMNLP 2019)

Le Fang[†], Chunyuan Li[§], Jianfeng Gao[§], Wen Dong[†], Changyou Chen[†]

Results:

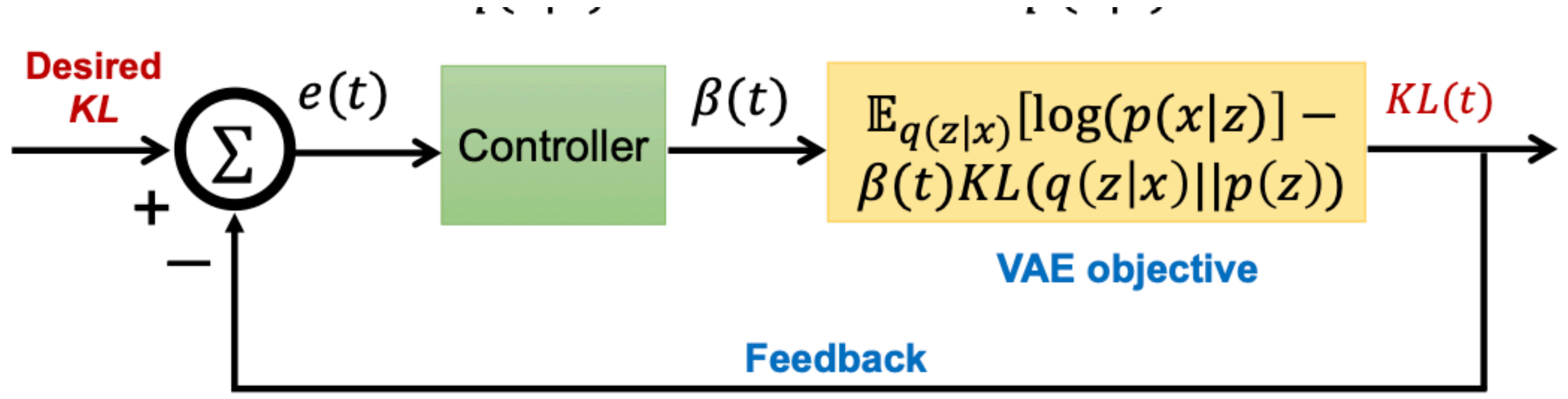
Methods	-ELBO↓	PPL↓	KL↑	MI↑	AU↑
		Dataset: PTB			
VAE	102.6	108.26	1.08	0.8	2
$\beta(0.5)$-VAE	104.5	117.92	**7.50**	3.1	5
SA-VAE	102.6	107.71	1.23	0.7	2
Cyc-VAE	103.1	110.50	3.48	1.8	5
iVAE	**87.6**	**54.46**	6.32	**3.5**	**32**
iVAE_{MI}	**87.2**	**53.44**	**12.51**	**12.2**	**32**
		Dataset: Yahoo			
VAE	328.6	61.21	0.0	0.0	0
$\beta(0.4)$-VAE	328.7	61.29	6.3	2.8	8
SA-VAE	327.2	60.15	5.2	2.7	10
Lag-VAE	326.7	59.77	5.7	2.9	15
iVAE	**309.5**	**48.22**	**8.0**	**4.4**	**32**
iVAE_{MI}	**309.1**	**47.93**	**11.4**	**10.7**	**32**
		Dataset: Yelp			
VAE	357.9	40.56	0.0	0.0	0
$\beta(0.4)$-VAE	358.2	40.69	4.2	2.0	4
SA-VAE	355.9	39.73	2.8	1.7	8
Lag-VAE	355.9	39.73	3.8	2.4	11
iVAE	**348.2**	**36.70**	**7.6**	**4.6**	**32**
iVAE_{MI}	**348.7**	**36.88**	**11.6**	**11.0**	**32**

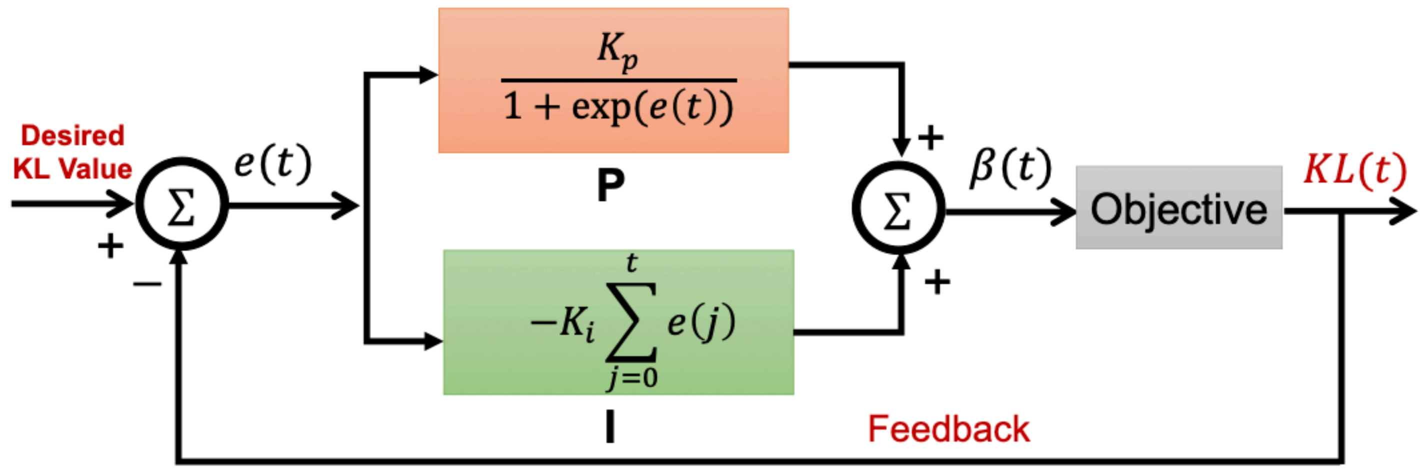
ControlVAE: Controllable Variational Autoencoder (ICML 2020)

Huajie Shao et al

Core ideas:

- Enhance β-VAE by controlling β with a new non-linear PI controller, β(t). The controller is not learnable, with Kp, Ki, Desired KL being hyperparameters.





- Intuition: Comparing error between Desired KL and Sampled KL (from inference output); if Sampled KL is too small, β(t) will be small so KL-divergence is encouraged to grow, and vice versa.

ControlVAE: Controllable Variational Autoencoder (ICML 2020)

Huajie Shao et al

Core ideas:

- PI Algorithm

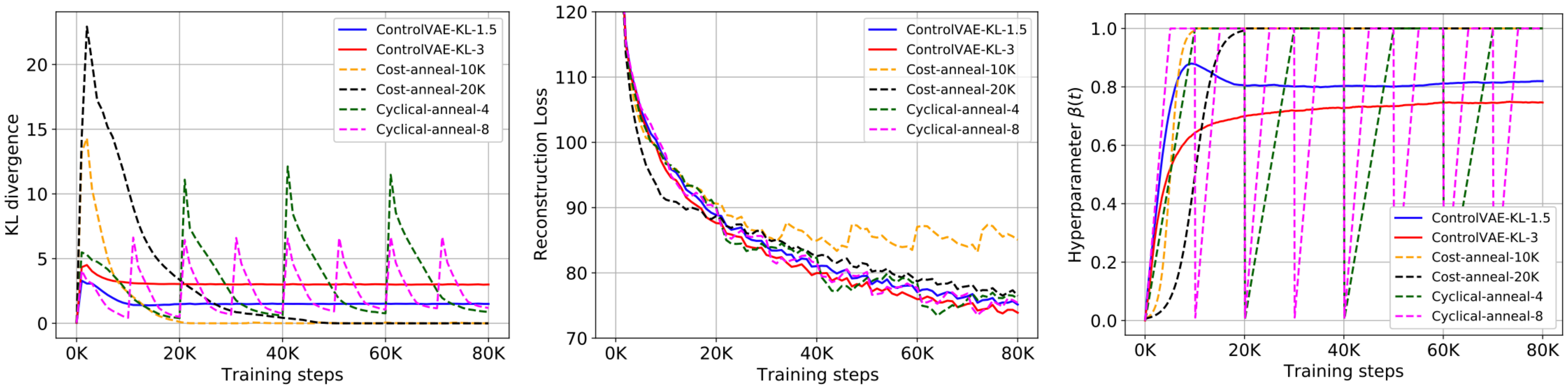
Algorithm 1 PI algorithm.

Input: desired KL v_{kl}, coefficients K_p, K_i, max/min value β_{max}, β_{min}, iterations N
Output: hyperparameter $\beta(t)$ at training step t
Initialization: $I(0) = 0$, $\beta(0) = 0$
for $t = 1$ **to** N **do**
  Sample KL-divergence, $\hat{v}_{kl}(t)$
  $e(t) \leftarrow v_{kl} - \hat{v}_{kl}(t)$
  $P(t) \leftarrow \frac{K_p}{1+\exp(e(t))}$
  if $\beta_{min} \leq \beta(t-1) \leq \beta_{max}$ **then**
    $I(t) \leftarrow I(t-1) - K_i e(t)$
  else
    $I(t) = I(t-1)$ // Anti-windup
  end if
  $\beta(t) = P(t) + I(t) + \beta_{min}$
  if $\beta(t) > \beta_{max}$ **then**
    $\beta(t) = \beta_{max}$
  end if
  if $\beta(t) < \beta_{min}$ **then**
    $\beta(t) = \beta_{min}$
  end if
  Return $\beta(t)$
end for

ControlVAE: Controllable Variational Autoencoder (ICML 2020)

Huajie Shao et al

Results:



(a) KL divergence (b) Reconstruction loss (c) $\beta(t)$

Methods/metric	Dis-1	Dis-2	self-BLEU-2	self-BLEU-3	PPL
ControlVAE-KL-35	**6.27K** $\pm$ 41	**95.86K** $\pm$ 1.02K	**0.663** $\pm$ 0.012	**0.447** $\pm$ 0.013	**8.81** $\pm$ 0.05
ControlVAE-KL-25	6.10K $\pm$ 60	83.15K $\pm$ 4.00K	0.698 $\pm$ 0.006	0.495 $\pm$ 0.014	12.47 $\pm$ 0.07
Cost anneal-KL-17	5.71K $\pm$ 87	69.60K $\pm$ 1.53K	0.721 $\pm$ 0.010	0.536 $\pm$ 0.008	16.82 $\pm$ 0.11
Cyclical (KL = 21.5)	5.79K $\pm$ 81	71.63K $\pm$ 2.04K	0.710 $\pm$ 0.007	0.524 $\pm$ 0.008	17.81 $\pm$ 0.33

FlowPrior: Learning Expressive Priors for Latent Variable Sentence Models (NAACL 2021)

Xiaoan Ding, Kevin Gimpel

Core ideas:

- Using normalizing flows (NF) for the prior distribution. In this paper, using *real-valued non-volume preserving* (real NVP) transformations (Dinh et al., 2016)
  - $z_L = f_L \circ f_{L-1} \circ \ldots \circ f_1(z_0)$, $z_0 \sim N(0, I)$, z_L is the sentence latent variable.

$$z_0 = f_1^{-1} \circ \ldots \circ f_{L-1}^{-1} \circ f_L^{-1}(z_L) \tag{5}$$

$$\log p_\psi(z_L) = \log p_0(z_0) - \sum_{l=1}^{L} \log |\det(\frac{\partial f_l(z_{l-1})}{\partial z_{l-1}})| \tag{6}$$

  - Inspired by Real-NVP, the choice of f is an affine transformation, which is f: $z_{i-1} \rightarrow z_i$

$$z_i^{(1:d)} = z_{i-1}^{(1:d)}$$
$$z_i^{(d+1:D)} = z_i^{(d+1:D)} \odot \exp(s(z_{i-1}^{(1:d)})) + t(z_{i-1}^{(1:d)})$$

    - 1) easily invertible
    - 2) its Jacobian determinant is easy to compute. (Do not require inverse/Jacobian of s and t)
  - So we can model s and t using NNs, denoted by $p_\psi(z)$

$$\mathbf{J} = \begin{bmatrix} \mathbb{I}_d & \mathbf{0}_{d\times(D-d)} \\ \frac{\partial \mathbf{y}_{d+1:D}}{\partial \mathbf{x}_{1:d}} & \mathrm{diag}(\exp(s(\mathbf{x}_{1:d}))) \end{bmatrix}$$

FlowPrior: Learning Expressive Priors for Latent Variable Sentence Models (NAACL 2021)

Xiaoan Ding, Kevin Gimpel

Core ideas:

- New Objective: Using importance weighting + Monte Carlo for KL

$$\mathcal{L}(\theta, \phi, \psi; x) = \log \frac{1}{N} \sum_{i=1}^{N} \frac{p_\theta(x|z^{(i)}) p_\psi(z^{(i)})}{q_\phi(z^{(i)}|x)} + \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x|z^{(i)}) - \mathrm{KL}_{\phi,\psi}(x, \{z^{(i)}\}_{i=1}^{N}) \quad \text{s.t. } z^{(i)} \sim q_\phi(z|x) \tag{7}$$

FlowPrior: Learning Expressive Priors for Latent Variable Sentence Models (NAACL 2021)

Xiaoan Ding, Kevin Gimpel

Core ideas:

- Training Scheme

1. Draw N samples $z_L^{(1)}, z_L^{(2)}, \ldots, z_L^{(N)}$ from the inference network using the reparameterization trick.
2. Perform the inverse transformation to get the image of each point under the base distribution: $z_0^{(1)}, z_0^{(2)}, \ldots, z_0^{(N)}$.
3. Compute the exact log likelihood of the sample prior with change of variable theorem (Eq. 6).
4. Compute and backpropagate the loss (Eq. 7).

FlowPrior: Learning Expressive Priors for Latent Variable Sentence Models (NAACL 2021)

Xiaoan Ding, Kevin Gimpel

Results:

Model	PPL(↓)	Recon(↓)	KL	AU(↑)	MI(↑)
VAE	101.40	101.28	0.00	0	0.00
Cyc-VAE	107.73	101.17	2.01	5	1.24
Lag-VAE	100.25	100.41	1.04	3	0.79
VAE + FB	101.56	99.84	4.46	32	0.90
Pre-VAE + FB	96.35	94.52	8.15	32	6.30
MoG-VAE	98.22	100.54	0.00	0	0.00
MoG-VAE + FB	97.50	99.44	2.35	32	0.68
Vamp-VAE	98.27	100.56	0.00	0	0.00
Vamp-VAE + FB	97.83	99.53	2.31	32	0.72
FlowPrior	94.72	98.46	3.28	2	2.25
FlowPrior + FB	93.58	99.20	7.21	31	2.83

Table 1: Language modeling results on PTB dataset.

Adversarial Poincaré Variational Autoencoder (APo- VAE) (NAACL 2021)

Shuyang Dai Zhe Gan Yu Cheng Chenyang Tao Lawrence Carin Jingjing Liu

Core ideas:

- Natural languages have latent hierarchy, but prior distribution from Euclidean space couldn’t capture it, so resorting to Riemannian Geometry.
- Proposing a prior based on Poincaré Ball Model in Hyperbolic space, with the following advantages:

$$\mathbb{B}_c^n := \{\boldsymbol{z} \in \mathbb{R}^n \mid c\|\boldsymbol{z}\|^2 < 1\}$$

  - It implies significant representation ability.
  - It’s a well defined compact metric space, with well defined algebraic operations that allows back-propagation.

$$\boldsymbol{z} \oplus_c \boldsymbol{z}' := \frac{(1 + 2c\langle \boldsymbol{z}, \boldsymbol{z}'\rangle + c\|\boldsymbol{z}'\|^2)\boldsymbol{z} + (1 - c\|\boldsymbol{z}\|^2)\boldsymbol{z}'}{1 + 2c\langle \boldsymbol{z}, \boldsymbol{z}'\rangle + c^2\|\boldsymbol{z}\|^2\|\boldsymbol{z}'\|^2}. \tag{4}$$

$$\exp_{\boldsymbol{\mu}}^c(\boldsymbol{u}) := \boldsymbol{\mu} \oplus_c (\tanh(\sqrt{c}\frac{\lambda_{\boldsymbol{\mu}}^c\|\boldsymbol{u}\|}{2})\frac{\boldsymbol{u}}{\sqrt{c}\|\boldsymbol{u}\|}),$$

$$\log_{\boldsymbol{\mu}}^c(\boldsymbol{y}) := \frac{2}{\sqrt{c}\lambda_{\boldsymbol{\mu}}^c}\tanh^{-1}(\sqrt{c}\|\boldsymbol{\kappa}_{\boldsymbol{\mu},\boldsymbol{y}}\|)\frac{\boldsymbol{\kappa}_{\boldsymbol{\mu},\boldsymbol{y}}}{\|\boldsymbol{\kappa}_{\boldsymbol{\mu},\boldsymbol{y}}\|}, \tag{5}$$

$$P_{\boldsymbol{0}\to\boldsymbol{\mu}}^c(\boldsymbol{v}) = \log_{\boldsymbol{\mu}}^c(\boldsymbol{\mu} \oplus_c \exp_{\boldsymbol{0}}^c(\boldsymbol{v})) = \frac{\lambda_{\boldsymbol{0}}^c}{\lambda_{\boldsymbol{\mu}}^c}\boldsymbol{v}. \tag{6}$$

where $\boldsymbol{\kappa}_{\boldsymbol{\mu},\boldsymbol{y}} := (-\boldsymbol{\mu}) \oplus_c \boldsymbol{y}$.

Adversarial Poincaré Variational Autoencoder (APo- VAE) (NAACL 2021)

Shuyang Dai Zhe Gan Yu Cheng Chenyang Tao Lawrence Carin Jingjing Liu

Core ideas:

- How does the defined compact metric space contribute to the formulation of a prior?
    - Poincaré ball based normal prior z ~ $N_{B^n_c}(\mu, \Sigma)$:

$$\boldsymbol{z} = \exp^c_{\boldsymbol{\mu}}\left(\frac{\lambda^c_{\mathbf{0}}}{\lambda^c_{\boldsymbol{\mu}}}\boldsymbol{v}\right), \boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}). \qquad (7)$$

    - Inspired by iVAE, we can enhance $\boldsymbol{v} := G(\boldsymbol{x}, \boldsymbol{\xi}; \phi_1)$ and $\boldsymbol{\mu} := F(\boldsymbol{x}; \phi_2)$, as outputs of encoder (ϕ). Then we get z from the above formula.
- Same as iVAE, they use dual form to evaluate KL(q_ϕ(z|x) ‖ p(z)):

$$\mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z})) = \max_{\boldsymbol{\psi}} \qquad (9)$$
$$\left\{\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\nu_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z}) - \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})} \exp \nu_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z})\right\},$$

- Same as iVAE, νψ is optimized by the following:

$$\mathcal{L}_1 = \mathbb{E}_{\boldsymbol{x}\sim \boldsymbol{X}}[\,\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\nu_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z}) - \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})} \exp \nu_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z})\,], \qquad (10)$$

Adversarial Poincaré Variational Autoencoder (APo- VAE) (NAACL 2021)

Shuyang Dai Zhe Gan Yu Cheng Chenyang Tao Lawrence Carin Jingjing Liu

Core ideas:

- Different from iVAE, here the real prior is not assumed Gaussian. It's estimated by sampling scheme used in VampPrior (Tomczak and Welling, 2018)

$$p_{\boldsymbol{\delta}}(\boldsymbol{z}) = \frac{1}{K}\sum_{k=1}^{K} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{s}_k), \qquad (12)$$

- $\delta := \{s_k\}^K_{k=1}$ is now learnable pseudo inputs. Replacing p(z) with $p_\delta(z)$ seeks to match the aggregated posterior $q(z) = \Sigma\ q_\phi(z|x_i) / N$.

Adversarial Poincaré Variational Autoencoder (APo- VAE) (NAACL 2021)

Shuyang Dai Zhe Gan Yu Cheng Chenyang Tao Lawrence Carin Jingjing Liu

Core ideas:

- For the geometry-aware decoder, they use a deterministic (and learnable) hyperbolic linear function f to extract feature, then pass to LSTM decoder. (a, b, together with the LSTM, are trainable parameters of decoder θ)

$$f^c_{\boldsymbol{a},\boldsymbol{b}}(\boldsymbol{z}) = \mathrm{sign}(\langle \boldsymbol{a}, \log^c_{\boldsymbol{b}}(\boldsymbol{z})\rangle_{\boldsymbol{b}}) \|\boldsymbol{a}\|_{\boldsymbol{b}}\, d^{\mathbb{B}}_c(\boldsymbol{z}, H^c_{\boldsymbol{a},\boldsymbol{b}}), \tag{8}$$

$$\text{where } H^c_{\boldsymbol{a},\boldsymbol{b}} = \{\boldsymbol{z} \in \mathbb{B}^n_c | \langle \boldsymbol{a}, \log^c_{\boldsymbol{b}}(\boldsymbol{z})\rangle_{\boldsymbol{b}} = 0\},$$

$$d^{\mathbb{B}}_c(\boldsymbol{z}, H^c_{\boldsymbol{a},\boldsymbol{b}}) = \frac{1}{\sqrt{c}} \sinh^{-1}\left(\frac{2\sqrt{c}|\langle \boldsymbol{\kappa}_{\boldsymbol{b},\boldsymbol{z}}, \boldsymbol{a}\rangle|}{(1-c\|\boldsymbol{\kappa}_{\boldsymbol{b},\boldsymbol{z}}\|^2)\|\boldsymbol{a}\|}\right)$$

- θ and ϕ are optimized by the following objective

$$\mathcal{L}_2 = \mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{X}} \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) - \nu_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{z})\big]. \tag{11}$$

Adversarial Poincaré Variational Autoencoder (APo- VAE) (NAACL 2021)

Shuyang Dai Zhe Gan Yu Cheng Chenyang Tao Lawrence Carin Jingjing Liu

Core ideas:

- Training Procedure

Algorithm 1 Training procedure of APo-VAE.

Input: Data samples $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$, Poincaré curvature c, and number of pseudo-input K.
Initialize $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\boldsymbol{\psi}$, and $\boldsymbol{\delta}$.
for $iter$ from 1 to max_iter **do**
    Sample a mini-batch $\{\boldsymbol{x}_m\}_{m=1}^{M}$ from $\boldsymbol{X}$ of size M.
    # Sampling in the Hyperbolic Space.
    Obtain $\boldsymbol{\mu}_m$ and $\boldsymbol{v}_m$ from $\text{EncNet}_{\boldsymbol{\phi}}(\boldsymbol{x}_m)$.
    Move $\boldsymbol{v}_m$ to $\boldsymbol{u}_m = P^c_{\mathbf{0}\to\boldsymbol{\mu}_m}(\boldsymbol{v}_m)$ by (6).
    Map $\boldsymbol{u}_m$ to $\boldsymbol{z}_m = \exp^c_{\boldsymbol{\mu}_m}(\boldsymbol{u}_m)$ by (5).
    # Update the dual function and the pseudo-input.
    Sample $\tilde{\boldsymbol{z}}_m$ by (12).
    Update $\boldsymbol{\psi}$ and $\boldsymbol{\delta}$ by gradient ascent on (10)
    # Update the encoder and decoder networks.
    Update $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ by gradient ascent on (11).
end for

Adversarial Poincaré Variational Autoencoder (APo- VAE) (NAACL 2021)

Shuyang Dai Zhe Gan Yu Cheng Chenyang Tao Lawrence Carin Jingjing Liu

Results:

Model	-ELBO	PPL	KL	MI	AU
	PTB				
VAE	102.6	108.26	1.1	0.8	2
β-VAE	104.5	117.92	7.5	3.1	5
SA-VAE	102.6	107.71	1.2	0.7	2
vMF-VAE	95.8	93.70	2.9	3.2	21
$\mathcal{P}$-VAE	91.4	76.13	4.5	2.9	23
iVAE	87.2	53.44	**12.5**	**12.2**	**32**
APo-VAE	87.2	53.32	8.4	4.8	**32**
APo-VAE+VP	**87.0**	**53.02**	8.9	4.5	**32**
	Yahoo				
VAE	328.6	61.21	0.0	0.0	0
β-VAE	328.7	61.29	6.3	2.8	8
SA-VAE	327.2	60.15	5.2	2.9	10
LAG-VAE	326.7	59.77	5.7	2.9	15
vMF-VAE	318.5	53.92	6.3	3.7	23
$\mathcal{P}$-VAE	313.4	50.57	7.2	3.3	27
iVAE	309.1	47.93	**11.4**	**10.7**	**32**
APo-VAE	286.2	47.00	6.9	4.1	**32**
APo-VAE+VP	**285.6**	**46.61**	8.1	4.9	**32**
	Yelp				
VAE	357.9	40.56	0.0	0.0	0
β-VAE	358.2	40.69	4.2	2.0	4
SA-VAE	357.8	40.51	2.8	1.7	8
LAG-VAE	355.9	39.73	3.8	2.4	11
vMF-VAE	356.2	51.03	4.1	3.9	13
$\mathcal{P}$-VAE	355.4	50.64	4.3	4.8	19
iVAE	348.7	36.88	11.6	**11.0**	**32**
APo-VAE	319.7	34.10	12.1	7.5	**32**
APo-VAE+VP	**316.4**	**32.91**	**12.7**	6.2	**32**