LREC-COLING 2024

Scale-VAE: Preventing Posterior Collapse in Variational Autoencoder

Tianbao Song[1], Jingbo Sun[2], Xin Liu[3], Weiming Peng[2]

1 School of Computer and Artificial Intelligence, Beijing Technology and Business University, China

2 School of Artificial Intelligence, Beijing Normal University, China

3 The 15th Research Institute of China Electronics Technology Group Corporation, China

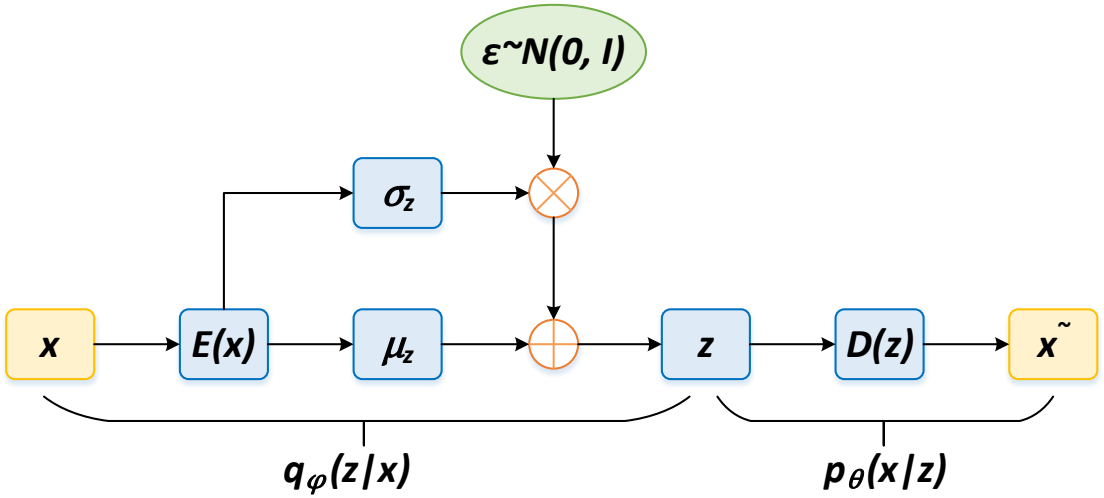
Posterior collapse in variational autoencoder

- Variational autoencoder [Kingma and Welling, 2014; Rezende et al., 2014]:
  - A widely used generative model.
  - Great capability in density estimation and representation learning.

- A notorious problem:
  - Fails to diversify the posteriors of different data instances.
  - Using a posterior that is almost identical to the prior for all data instances.
  - Particularly when employing the strong autoregressive generation networks.
  - Termed posterior collapse or KL vanishing.

Causes of the posterior collapse problem

- The KL term is an upper bound on the amount of the transmitted information from the input data.
- There is a contradiction between the reconstruction term and the KL term.
- The information in latent variables is difficult to exploit, especially in the early stages of training.



$$L_{ELBO} = E_{p(X)}[E_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}[q_\phi(z|x)\|p(z)]]$$

reconstruction term

KL term

Solutions to the posterior collapse problem

- Weaken the decoder, forcing the model to rely on latent variables.
  - E.g., CNN as decoder, word tokens dropout, fraternal dropout on decoder, ……
- Alleviate the contradiction between the reconstruction term and the KL term.
  - E.g., KL annealing, cyclical annealing, β-VAE, Free-Bits, ……
- Enforce the relation between latent variables and input data by MI-based terms.
  - E.g., Fisher autoencoder, KL term between the aggregated posterior and prior, mutual posterior-divergence regularization, ……
- Reduce the difficulty of exploiting latent variables.
  - E.g., Skip-VAE, SA-VAE, Agg-VAE, BN-VAE, ……

Limitations of previous work

- Do not make full use of the expressive generation network.
- Have non-smooth optimizations or are time consuming.
- Additional optimization methods are required due to the intractability of the added regularization term in the objective.
- The relative scale of the regularization term and original objective requires deliberate tuning.
- Lead to semantic confusion in the latent space.

The proposed method: Scale-VAE

- Encoder:

$$Enc_{\varphi}(x) = q_{\varphi}(z|x) = N(\mu_x, \sigma_x^2)$$

$$\mu_x = (\mu_{x,1}, \mu_{x,2}, \dots, \mu_{x,d}, \dots, \mu_{x,n})$$

- The scale-up factor:

$$f = (f_1, f_2, \dots, f_d, \dots, f_n)$$

$$f_d = \frac{des\_std}{Std[\mu_{X,d}]}$$

- The scaled-up posterior distribution:

$$\hat{q}_{\varphi}(z|x) = N(\hat{\mu}_x, \sigma_x^2)$$

$$\hat{\mu}_x = (\hat{\mu}_{x,1}, \hat{\mu}_{x,2}, \dots, \hat{\mu}_{x,d}, \dots, \hat{\mu}_{x,n})$$

$$\hat{\mu}_{x,d} = f_d \cdot \mu_{x,d}$$

- Objective:

$$L_{scale} = E_{p(X)}[E_{\hat{q}_{\varphi}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\theta}(x|z)] - D_{KL}[q_{\varphi}(z|x)||p(z)]]$$

The proposed method: Scale-VAE

- Objective:

$$L_{scale} = E_{p(X)}[E_{\hat{\boldsymbol{q}}_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(x|z)] - D_{KL}[q_\varphi(z|x)||p(z)]]$$

  - Keep $Std[\hat{\mu}_{X,d}]$ around des_std.
    - Distinguish the latent variables of different data instances and obtain effective information from them more easily.
  - $\hat{q}_\varphi(z|x)$ is used in the reconstruction term but $q_\varphi(z|x)$ is used in the KL term.
    - Avoid pushing the posterior $q_\varphi(z|x)$ even further into the prior $p(z)$.
    - Make the whole objective reach the optimum instead of forcing the KL term to be larger than a certain positive constant.
    - Using $f \cdot z, z{\sim}p(z)$ during inference stage that is consistent with training; The latent space used for generation is smooth without many discontinuous holes due to large $Std[\hat{\mu}_{X,d}]$.

The proposed method: Scale-VAE

- Challenge:
  - The factor f can only be computed within a mini-batch.
  - f used in each mini-batch are different and will cause clutter in the latent space.
- Solution:
  - Each mini-batch uses its own factor f only in the initial f_epo training epochs.
  - Thereafter, record f of all the mini-batches in each training epoch and take the average $\bar{f}$ as the factor of the next training epoch.

Algorithm 1 Training Procedure of Scale-VAE

Initialize ϕ, θ, des_std and f_epo
$i \leftarrow 1$
while not convergence **do**
  for $\mathbf{x}$ in mini-batches **do**
    $\mu_{\mathbf{x}}, \sigma^2_{\mathbf{x}} = Enc_{\phi}(\mathbf{x})$
    $f = des\_std / Std[\mu_{\mathbf{x}}]$
    if $i \leq f\_epo$ **then**
      $\hat{\mu}_{\mathbf{x}} = f \cdot \mu_{\mathbf{x}}$
    else
      $\hat{\mu}_{\mathbf{x}} = \bar{f} \cdot \mu_{\mathbf{x}}$
    end if
    Sample $z \sim N(\hat{\mu}_{\mathbf{x}}, \sigma^2_{\mathbf{x}})$
    Generate $\mathbf{x}$ from $Dec_{\theta}(z)$
    $g_{\phi,\theta} \leftarrow -\nabla_{\phi,\theta} L_{scale}(\mathbf{x}; \phi, \theta)$
    Update ϕ, θ according to $g_{\phi,\theta}$
  end for
  $\bar{f} = Average(f)$
  $i \leftarrow i + 1$
end while

Connections with previous work

- Equivalent to increasing the mutual posterior divergence (MPD), that is, increasing the divergence among the posterior distribution family for better distinguish.
- Equivalent to increasing the mutual information (MI) between latent variables and input data.

$$
\begin{aligned}
2MPD &= 2E_{p(X)}[D_{SKL}[q_\phi(z|x_1)\|q_\phi(z|x_2)]] \\
&= \sum_{d=1}^{n} E_{p(X)}[\frac{(\mu_{x_1,d}-\mu_{x_2,d})^2}{\sigma_{x_1,d}^2}]+ \\
&\quad \sum_{d=1}^{n} E_{p(X)}[\sigma_{x,d}^2]E_{p(X)}[\frac{1}{\sigma_{x,d}^2}]-1
\end{aligned}
$$

$$
MPD \geq \frac{1}{C}\sum_{d=1}^{n} Var_{p(X)}[\mu_{x,d}],\ \text{if } \sigma_{x,d}^2 \leq C
$$

$$
\begin{aligned}
MPD &= E_{p(X)}[D_{KL}[q_\phi(z|x)\|q_\phi(z)] \\
&\quad + D_{KL}[q_\phi(z)\|q_\phi(z|x)]]
\end{aligned}
$$

$$
\begin{aligned}
I_q(x;z) &= E_{p(X)}[D_{KL}[q_\phi(z|x)\|p(z)]] \\
&\quad - D_{KL}[q_\phi(z)\|p(z)] \\
&= E_{p(X)}[D_{KL}[q_\phi(z|x)\|q_\phi(z)]]
\end{aligned}
$$

Connections with previous work

- Batch normalization-based method [Zhu et al., 2020; Shen et al., 2021]:
  - Regularizes $\mu_{x,d}$ with batch normalization. $\hat{\mu}_{x,d} = \gamma \frac{\mu_{x,d} - \mu_{Bd}}{\sigma_{Bd}} + \beta$
  - The distribution of $\mu_{x,d}$ has the mean of β and the standard deviation of γ, and $E[KL] \geq n \cdot (\gamma^2 + \beta^2)/2$.
- Differences with batch normalization-based method:
  - The motivation is not to force the KL term to be larger than a certain positive constant.
  - The model can feel free to optimize the KL term because it is still computed using the original posteriors, and the scaling-up will help the model make better use of the latent variables.
  - All data instances use the same scale-up factor, and the problem of latent space clutter in the batch normalization-based method does not occur.

Experiments-Density estimation

Model	Yahoo				Yelp			
	NLL	KL	MI	AU	NLL	KL	MI	AU
LSTM-LM	328.0	-	-	-	357.5	-	-	-
VAE	328.6	0.2	0.2	0.8	358.0	0.1	0.1	0.2
cyclical	330.5	2.1	2.1	2.3	359.3	2.1	2.0	4.2
β-VAE(0.4)	328.4	7.7	7.1	7.3	358.0	5.4	5.1	3.6
Free-Bits(0.1)	328.3	3.4	2.4	32.0	357.0	4.8	2.6	32.0
δ-VAE(0.1)	329.7	3.2	0.0	2.0	357.9	3.2	0.0	0.0
SA-VAE$*$	327.2	5.2	3.7	9.8	355.9	2.8	1.7	8.4
Skip-VAE$*$	328.5	2.3	1.3	8.1	357.6	1.9	1.0	7.4
Agg-VAE$*$	326.7	5.7	2.9	15.0	355.9	3.8	2.4	11.3
MAE$*$(1, 0.2|2, 0.2)	332.1	5.8	3.5	28.0	362.8	8.0	4.6	32.0
BN-VAE(0.6)	326.9	6.5	5.8	32.0	356.6	6.5	5.7	32.0
BN-VAE(0.0, 0.7)	331.3	7.8	0.0	0.0	360.4	7.8	0.0	0.0
DU-VAE(0.6, 0.8|0.5, 0.8)	326.9	8.8	7.2	28.0	356.2	6.7	5.9	20.0
Scale-VAE(0.7, 1|0.7, 7)	325.0	7.1	8.3	32.0	353.7	5.4	8.2	32.0
Scale-VAE(0.9, 1|0.9, 7)	323.1	9.5	9.1	32.0	351.7	7.2	9.1	32.0
Scale-VAE(1.1, 1|1.1, 7)	**321.3**	8.4	9.2	32.0	**349.9**	6.7	9.2	32.0
Scale-VAE$\star$(0.7, 1|1.1, 7)	325.9	6.4	7.9	32.0	350.7	8.1	9.2	32.0

Table 1: Density estimation performance on Yahoo and Yelp. The results are the mean values across 5 different random runs. $*$ indicates the results are referred from Kim et al. (2018), He et al. (2018) and Shen et al. (2021). $\star$ indicates that KL annealing is not used. Hyperparameters are listed in brackets and split by | if different on different datasets.

- Dataset:
  - Yahoo
  - Yelp
- Metric:
  - NLL
  - KL
  - MI
  - AU

Experiments-Density estimation

Model	Yahoo		Yelp	
	Hours	Ratio	Hours	Ratio
VAE	3.50	1.00	4.90	1.00
BN-VAE	3.55	1.01	4.70	0.95
Scale-VAE	**3.22**	**0.91**	**4.29**	**0.87**

Table 2: Comparison of training time to convergence. Ratio indicates the relative ratio to VAE. The results are the mean values across 5 different random runs with the best parameters in Table 1 for each model.

Scale-VAE		Yahoo			
des_std	f_epo	NLL	KL	MI	AU
0.3	1	324.1	1.8	8.4	23.2
0.5	1	325.7	3.7	7.0	31.8
0.7	1	325.0	7.1	8.3	32.0
0.7	3	324.6	6.9	8.4	32.0
0.7	5	324.7	7.1	8.3	32.0
0.9	1	323.1	9.5	9.1	32.0
1.1	1	321.3	8.4	9.2	32.0
1.3	3	318.6	8.8	9.2	32.0
1.5	7	326.5	3.0	8.3	32.0

Table 3: Density estimation performance of Scale-VAE with different hyperparameters on Yahoo and Yelp. The results are the mean values across 5 different random runs.

Experiments-Representation learning

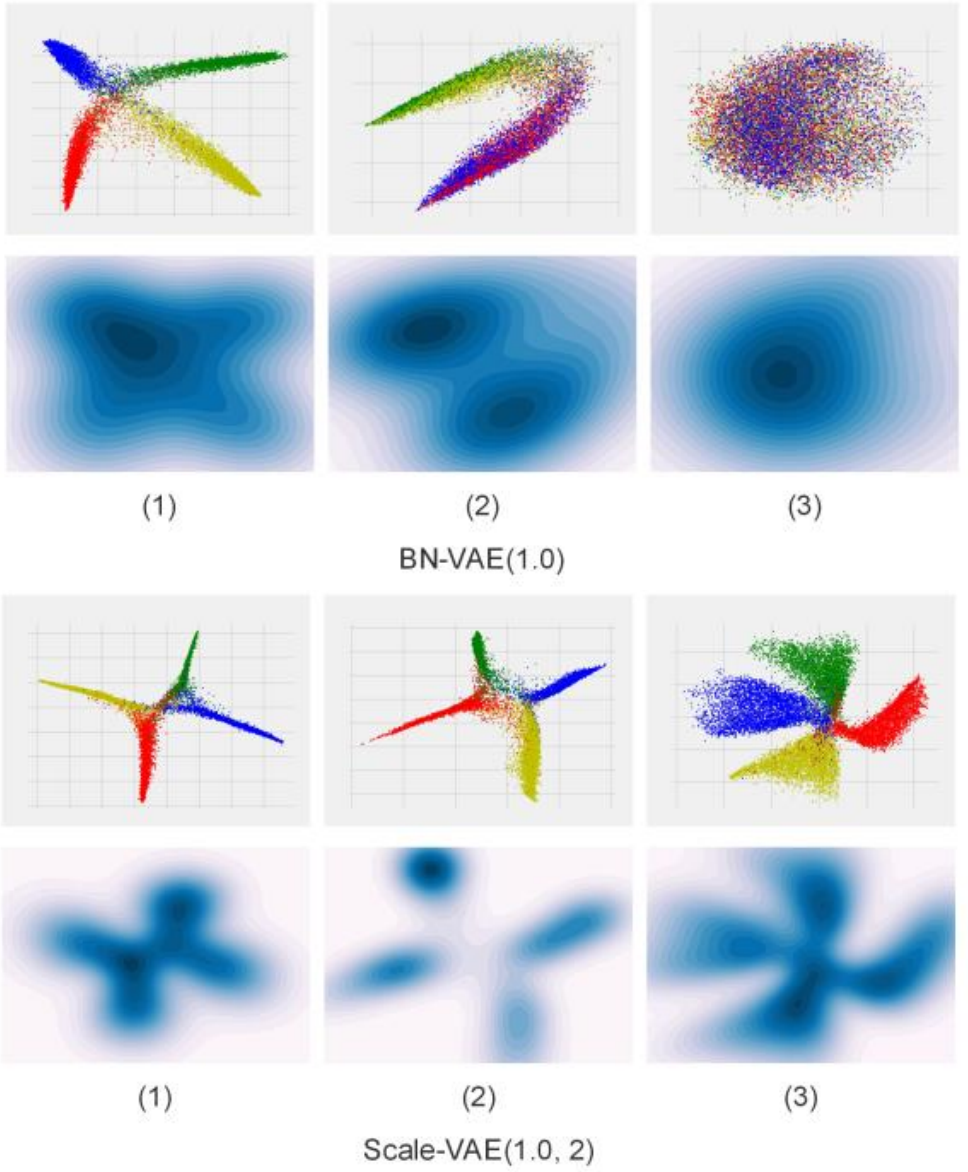
#labeled-data	100	500	1K	2K	10K
VAE	72.0	75.9	76.5	78.6	80.0
β-VAE(0.4)	82.0	83.7	84.3	84.8	86.2
FB(0.1)	72.0	75.9	76.5	78.6	80.0
δ-VAE(0.1)	58.9	59.8	60.5	59.7	61.2
Agg-VAE$*$	75.1	77.2	78.5	79.3	80.1
MAE$*$(2, 0.2)	61.5	61.7	62.4	63.6	63.7
BN-VAE(0.6)	85.4	88.7	89.8	90.2	90.4
DU-VAE(0.5, 0.8)	85.1	86.4	88.2	89.0	89.1
DU-VAE$*$(0.5, 0.8)	**88.9**	89.6	90.4	90.5	90.8
Scale-VAE(0.7, 1)	87.7	**89.8**	**90.7**	**91.3**	**91.2**

Table 4: Classification accuracy with different amounts of labeled data in Yelp. $*$ indicates the results are referred from Shen et al. (2021). Hyper-parameters are listed in brackets.

- Dataset:
  - Downsampled version of Yelp sentiment dataset
- Metric:
  - Training a one-layer linear classifier using the means of posterior distributions
  - Classification accuracy

Experiments-Latent space property

- Dataset: Synthetic dataset
- Three mini-batch partitioning modes (Mini-batch data distribution / Mini-batch splitting):
  - Random / Unchanged
  - Random / Resplit
  - Each mini-batch comes from the same Gaussian component / Unchanged



Experiments-Latent space property

Model	BN-VAE(0.6)	Scale-VAE(1.1, 1)
PPL	**260.44**	274.93
Gram-2	97.77	**98.08**
Gram-3	**87.24**	84.19
Gram-4	**68.10**	60.24
Entropy	5.72	**5.91**
Dist-1(E-02)	0.82	**1.33**
Dist-2(E-05)	0.90	**1.89**
Dist-3(E-05)	1.69	**4.40**

Table 5: Quality and diversity evaluation results of the generated sentences on Yahoo. Gram-2/3/4 denote the proportion of 2/3/4-grams in the generated sentences that appear in the Yahoo dataset. Dist-1/2/3 denote the proportion of different 1/2/3-grams in the generated sentences.

- Dataset:
  - Yahoo
- Metric:

  Quality
  - PPL
  - Gram-2/3/4

  Diversity
  - Entropy
  - Dist-1/2/3

Conclusions

- Scale-VAE to solve posterior collapse:
  - Motivated by reducing the difficulty of exploiting latent variables rather than forcing the KL term to be larger than a certain positive constant.
  - It alleviates the contradiction between the reconstruction term and the KL term, so that it can not only learn the latent space that is smooth and tends to the prior, but also improve the density estimation and representation learning.
  - It does not cause clutter in the latent space.