使用 Batch Normalization 防止变分自编码器中 KL 散度的消失

本论文由腾讯 AI Lab 主导，和佛罗里达大学合作完成。作者利用通过直接计算 KL 散度在数据集中的期望并使其有一个大于 0 的下界从而解决这个问题。作者基于此提出了 BN-VAE，在编码器的输出使用 batch normalization。在没有增加额外的训练参数和训练量的情况下有效缓解了 KL 消失的问题。

A Batch Normalized Inference Network Keeps the KL Vanishing Away

变分自编码器（VAE）是一种很常用的生成模型，它希望构建一个从隐变量空间到数据空间的映射。因为其可以从分布中采样，每次都有一定的随机性，所以在多样性文本生成中有一席之地。然而在文本生成中，decoder 一般为很强的自回归模型比如 RNN 家族 （LSTM，GRU 等）或者最近的 Transformer 结构。当 VAE 与他们配合使用时往往会产生 KL 散度消失的现象，因为 decoder 的自回归性，往往会忽略掉 VAE 中的隐变量部分。

之前已经有很多很好的工作来试图解决这个问题，但是都需要增加额外的参数或者训练过程。如何不增加训练负担并且有效地防止 KL 散度的消失是本文研究的动机。VAE 需要优化边际似然概率的下界，即 Evidence Lower Bound （ELBO）:

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

在我们实际运用 VAE 时，正态分布往往是一个通常的选择，从来上式中 KL 的

项可以由如下计算：

$$KL = \frac{1}{2}\sum_{i=1}^{n}(\mu_i^2 + \sigma_i^2 - \log\sigma_i^2 - 1)$$

式中变量为在隐空间的第 i 维的后验分布的均值和标准差。在实际计算中，我们往往会用到 batch 训练，所以上式在训练过程中可以进一步进行计算得到：

$$\begin{aligned} KL &= \frac{1}{2b}\sum_{j=1}^{b}\sum_{i=1}^{n}(\mu_{ij}^2 + \sigma_{ij}^2 - \log\sigma_{ij}^2 - 1) \\ &= \frac{1}{2}\sum_{i=1}^{n}(\frac{\sum_{j=1}^{b}\mu_{ij}^2}{b} + \frac{\sum_{j=1}^{b}\sigma_{ij}^2}{b} \\ &- \frac{\sum_{j=1}^{b}\log\sigma_{ij}^2}{b} - 1). \end{aligned}$$

当 batch size 很大时，上式中的 KL 项将会近似于整个数据集的 KL 的均值。由此，我们可以通过限制均值和方差的分布来限制 KL 在数据集中的分布。这样 KL 就相当于是一个关于隐变量的后验分布参数的分布。此外当 batch size 足够大时上式可以表示成如下：

$$\begin{aligned} \mathrm{E}[KL] &= \frac{1}{2}\sum_{i=1}^{n}(\mathrm{Var}[\mu_i] + \mathrm{E}^2[\mu_i] \\ &+ \mathrm{E}[\sigma_i^2] - \mathrm{E}[\log\sigma_i^2] - 1) \\ &\geq \frac{1}{2}\sum_{i=1}^{n}(\mathrm{Var}[\mu_i] + \mathrm{E}^2[\mu_i]) \end{aligned}$$

由于加号后的一项恒大于等于 0，所以不等式成立。通过这个变换不难想到可以使用 batch normalization 来对均值的分布进行约束。对后验分布中的均值进行如下操作：

$$\hat{\mu}_i = \gamma\frac{\mu_i - \mu_{\mathcal{B}i}}{\sigma_{\mathcal{B}i}} + \beta$$

式中 gamma 和 beta 为 batch normalization 中的参数，分别可以控制 mu

分布的方差和均值。将上式中的 mu 替换到 KL 的计算式子中我们可以得到：

$$\begin{aligned}\mathrm{E}[KL] &\geq \frac{1}{2}\sum_{i}^{n}(\mathrm{Var}[\mu_i] + \mathrm{E}^2[\mu_i]) \\ &= \frac{n\cdot(\gamma^2+\beta^2)}{2}.\end{aligned}$$

至此，我们可以通过更改 gamma 和 beta 参数来控制 KL 分布的期望的下界。

整体流程可以总结为：

Algorithm 1 BN-VAE training.

Initialize ϕ and $\boldsymbol{\theta}$.
for $i = 1, 2, \cdots$ Until Convergence **do**
  Sample a mini-batch $\mathbf{x}$.
  $\mu, \log\sigma^2 = f_\phi(\mathbf{x})$.
  $\mu' = BN_{\gamma,\beta}(\mu)$.
  Sample $\mathbf{z} \sim \mathcal{N}(\mu', \sigma^2)$ and reconstruct $\mathbf{x}$ from $f_\theta(\mathbf{z})$.
  Compute gradients $\mathbf{g}_{\phi,\theta} \leftarrow \nabla_{\phi,\theta}\mathcal{L}(\mathbf{x};\phi,\theta)$.
  Update ϕ, θ using $\mathbf{g}_{\phi,\theta}$.
end for

同样，我们可以将这个方法应用于 CVAE 中，具体证明过程在此不赘述。算法如下：

Algorithm 2 BN-CVAE training.

Initialize ϕ, θ and κ.
for $i = 1, 2, \cdots$ Until Convergence **do**
  Sample a mini-batch $\mathbf{x}, \mathbf{y}$.
  $\mu_q,\ \log\sigma_q^2 = f_\phi(\mathbf{x},\mathbf{y})$ and $\mu_p,\ \log\sigma_p^2 = f_\theta(\mathbf{x})$.
  $\mu_q' = BN_{\gamma,\beta}(\mu_q)$.
  Sample $\mathbf{z} \sim \mathcal{N}(\mu_q', \sigma_q^2)$ and reconstruct $\mathbf{y}$ from $f_\kappa(\mathbf{z},\mathbf{x})$.
  Compute gradients $\mathbf{g}_{\phi,\theta,\kappa} \leftarrow \nabla_{\phi,\theta,\kappa}\mathcal{L}'$.
  Update ϕ, θ, κ using $\mathbf{g}_{\phi,\theta,\kappa}$.
end for

为了验证 BN-VAE 方法的有效性我们进行了语言模型，用隐变量进行文本分类以及对话生成的实验。

	Yahoo				Yelp			
Model	**NLL**	**KL**	**MI**	**AU**	**NLL**	**KL**	**MI**	**AU**
Without a pretrained AE encoder								
CNN-VAE	≤332.1	10.0	-	-	≤359.1	7.6	-	-
LSTM-LM	328	-	-	-	351.1	-	-	-
VAE	328.6	0.0	0.0	0.0	357.9	0.0	0.0	0.0
β-VAE (0.4)	328.7	6.3	2.8	8.0	358.2	4.2	2.0	4.2
cyclic *	330.6	2.1	2.0	2.3	359.5	2.0	1.9	4.1
Skip-VAE *	328.5	2.3	1.3	8.1	357.6	1.9	1.0	7.4
SA-VAE	327.2	5.2	2.7	9.8	**355.9**	2.8	1.7	8.4
Agg-VAE	**326.7**	5.7	2.9	15.0	**355.9**	3.8	2.4	11.3
FB (4)	331.0	4.1	3.8	3.0	359.2	4.0	1.9	32.0
FB (5)	330.6	5.7	2.0	3.0	359.8	4.9	1.3	32.0
δ-VAE (0.1) *	330.7	3.2	0.0	0.0	359.8	3.2	0.0	0.0
vMF-VAE (13) *	327.4	2.0	-	32.0	357.5	2.0	-	32.0
BN-VAE (0.6) *	**326.7**	6.2	5.6	32.0	356.5	6.5	5.4	32.0
BN-VAE (0.7) *	327.4	8.8	7.4	32.0	**355.9**	9.1	7.4	32.0
With a pretrained AE encoder								
cyclic *	333.1	25.8	9.1	32.0	361.5	20.5	9.3	32.0
FB (4) *	**326.2**	8.1	6.8	32.0	356.0	7.6	6.6	32.0
δ-VAE (0.15) *	331.0	5.6	1.1	11.2	359.4	5.2	0.5	5.9
vMF-VAE (13) *	328.4	2.0	-	32.0	357.0	2.0	-	32.0
BN-VAE (0.6) *	326.7	6.4	5.8	32.0	**355.5**	6.6	5.9	32.0
BN-VAE (0.7) *	326.5	9.1	7.6	32.0	355.7	9.1	7.5	32.0

表一：在 Yahoo 和 Yelp 数据集上语言模型的结果。

	Yahoo		**Yelp**	
Model	**Hours**	**Ratio**	**Hours**	**Ratio**
VAE	3.83	1.00	4.50	1.00
SA-VAE	52.99	12.80	59.37	12.64
Agg VAE	11.76	2.84	21.44	4.56
AE+FB	7.70	2.01	9.22	2.05
BN-VAE	3.98	1.04	4.60	1.02

表二：在 Yahoo 和 Yelp 数据集上训练模型的时间。

从上面两张表中可以看出，BN-VAE 取得了很好的效果并且训练时间和 VAE 相差无几。

在用隐变量进行文本分类中 BN-VAE 同样表现十分出色，结果如下表。

#label	100	500	1k	2k	10k
AE	81.1	86.2	90.3	89.4	94.1
VAE	66.1	82.6	88.4	89.6	94.5
δ-VAE	61.8	61.9	62.6	62.9	93.8
Agg-VAE	80.9	85.9	88.8	90.6	93.7
cyclic	62.4	75.5	80.3	88.7	94.2
FB (9)	79.8	84.4	88.8	91.12	94.7
AE+FB (6)	87.6	90.2	92.0	93.4	94.9
BN-VAE (0.7)	**88.8**	**91.6**	**92.5**	**94.1**	**95.4**

表三：在 Yelp(采样) 数据集中的分类结果。

Topic: ETHICS IN GOVERNMENT		
Context: have trouble drawing lines as to what's illegal and what's not		
Target (statement): well i mean the other problem is that they're always up for		
CVAE	CVAE (BOW)	BN-CVAE
1. yeah	1. yeah	1. it's not a country
2. yeah	2. oh yeah they're not	2. it is the same thing that's what i think is about the state is a state
3. yeah	3. no it's not too bad	3. yeah it's

表四：不同算法下的采样回复。

在对话实验中，由于 BN-VAE 可以得到相对可控的 KL 值，使得采样出来的回答更加符合原文语义。样例如表 4。