电子科技大学
University of Electronic Science and Technology of China

Neural Variational Inference

Di Wu

Data Mining Lab, Big Data Research Center, UESTC
Email: wudi.araragi@qq.com
Homepage: http://dm.uestc.edu.cn

Begin with VAE

Variational auto-encoder is used to perform approximate inference on probabilistic models which have intractable posterior distribution over latent variables and parameters. It fits a approximate inference model (also called recognition model, just an encoder) to the true posterior using a estimator of the ELBO.

Two key points:

- reparameterization trick
- efficient optimization of ELBO by stochastic gradient

VAE Model

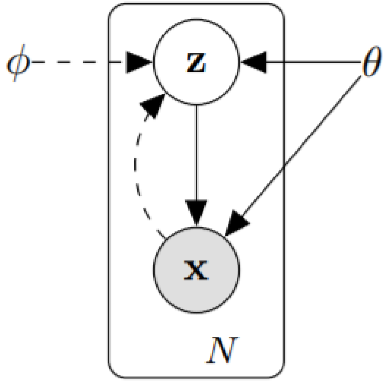


Figure 1: The type of directed graphical model under consideration. Solid lines denote the generative model $p_{\boldsymbol{\theta}}(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, dashed lines denote the variational approximation $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ to the intractable posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$. The variational parameters $\boldsymbol{\phi}$ are learned jointly with the generative model parameters $\boldsymbol{\theta}$.

Marginal log likelihood is $\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$

Rewrite the likelihood using a variational distribution $q_\phi(z|x)$:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$$

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) + \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\right]$$

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})\right] \qquad (*1*)$$

Reparameterization Trick: Estimator of the ELBO

We want to differentiate and optimize the lower bound $\mathcal{L}$ in (1) w.r.t ϕ and θ, the main difficulty is the gradient of ϕ. With a well chosen posterior $q_\phi(z|x)$, we can reparameterize variable $\tilde{z} \sim q_\phi(z|x)$ using a differentiable transformation $g_\phi(\epsilon, x), \epsilon$ is an auxiliary noise variable:

$$\widetilde{\mathbf{z}} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

Monte Carlo estimates:

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}\left[f(\mathbf{z})\right] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}^{(i)}))\right] \simeq \frac{1}{L}\sum_{l=1}^{L} f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i)})) \quad \text{where} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

The ELBO can be rewrited as:

$$\widetilde{\mathcal{L}}^{B}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L}(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}))$$

$$\text{where} \quad \mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

Algorithm 1 Minibatch version of the Auto-Encoding VB (AEVB) algorithm. Either of the two SGVB estimators in section 2.3 can be used. We use settings $M = 100$ and $L = 1$ in experiments.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset)
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from noise distribution $p(\boldsymbol{\epsilon})$
  $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (Gradients of minibatch estimator (8))
  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update parameters using gradients $\mathbf{g}$ (e.g. SGD or Adagrad [DHS10])
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$
return $\boldsymbol{\theta}, \boldsymbol{\phi}$

Flexibilty of the choice of the prior and the design of variational posterior

How to Apply VAE Framework to NLP

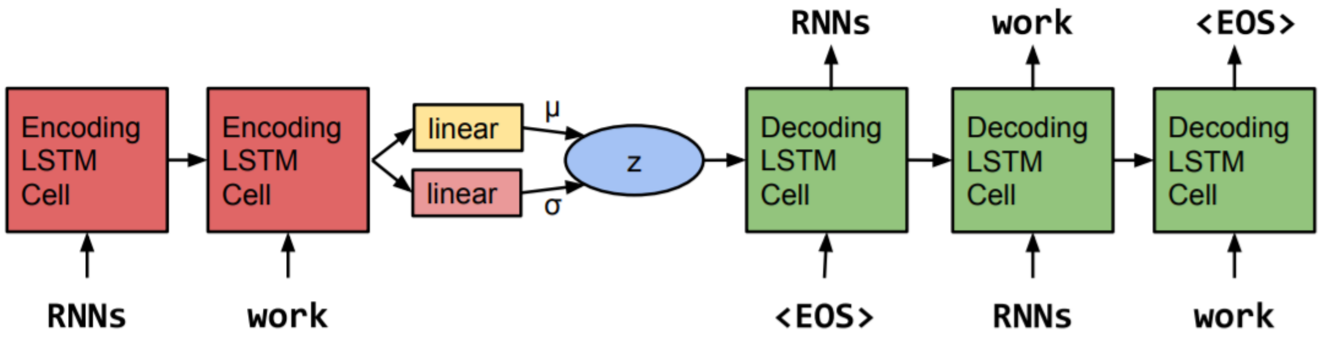


Figure 1: The core structure of our variational autoencoder language model. Words are represented using a learned randomly-initialized dictionary of embedding vectors. $\vec{z}$ is a vector-valued latent variable with a Gaussian prior and an approximate posterior parameterized by the encoder's outputs μ and σ. <EOS> marks the end of each sequence.

Simple approach:

$$\textit{prior} \quad p(z) = \mathcal{N}(\mu_0, \sigma_0^2)$$
$$\textit{posterior} \quad q(z|x) = \mathcal{N}(\mu = f(x), \sigma = g(x))$$

Variable Z can be seen as the sentence sematic(global feature, like topic)

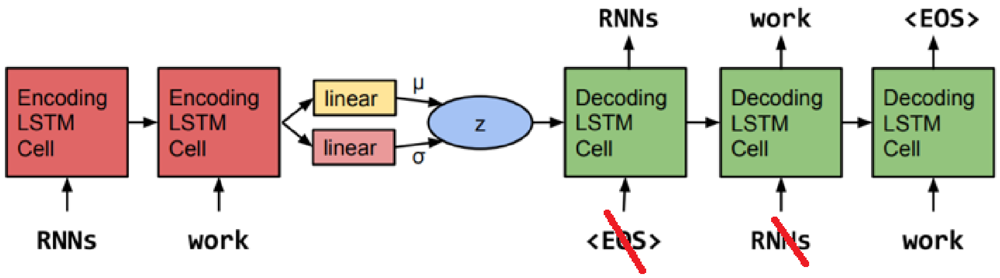
It seems okay, but:

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

Because RNN can express arbitrary distributions over the output sentences, so RNN can achieve optimal likelihood even without Z, so KL will fall down to zero, actually Z doesn't be learned

How to alleviate:

- Word dropout:



- KL cost annealing: $\mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})\right] - \mathrm{W}^* D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}))$

W increases gradually from 0

Neural Variational Inference for Text Processing(1)

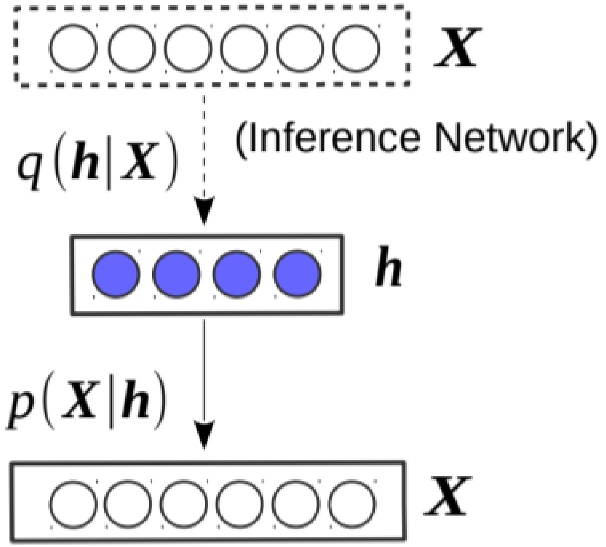


Figure 1. NVDM for document modelling.

都是套路：

X:document representaion(e.g. bag of words),x_i is the ith word(one hot)

h:continuous hidden variable which generate all the words independently

prior $p_\theta(h)$ is a Gaussian

$p_\theta(x_i|h) = \frac{exp\{E(x_i;h,\theta)\}}{\sum_{j=1}^{|V|} exp\{E(x_j;h,\theta)\}}$,where $E(x_i;h,\theta) = h^T R x_i - b_{x_i}, R \in \mathbb{R}^{K\times|V|}$

posterior $q_\phi(h|X) = \mathcal{N}(h|\mu(X), diag(\sigma^2(X)))$

$$\pi = g(f_X^{MLP}(X))$$
$$\mu = l_1(\pi), log\sigma = l_2(\pi)$$

$$\mathcal{L} = \mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{X})}\left[\sum_{i=1}^{N} \log p_\theta(\boldsymbol{x}_i|\boldsymbol{h})\right] - D_{\mathrm{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{X})\|p(\boldsymbol{h})]$$

ICML 2016

Neural Variational Inference for Text Processing(2)

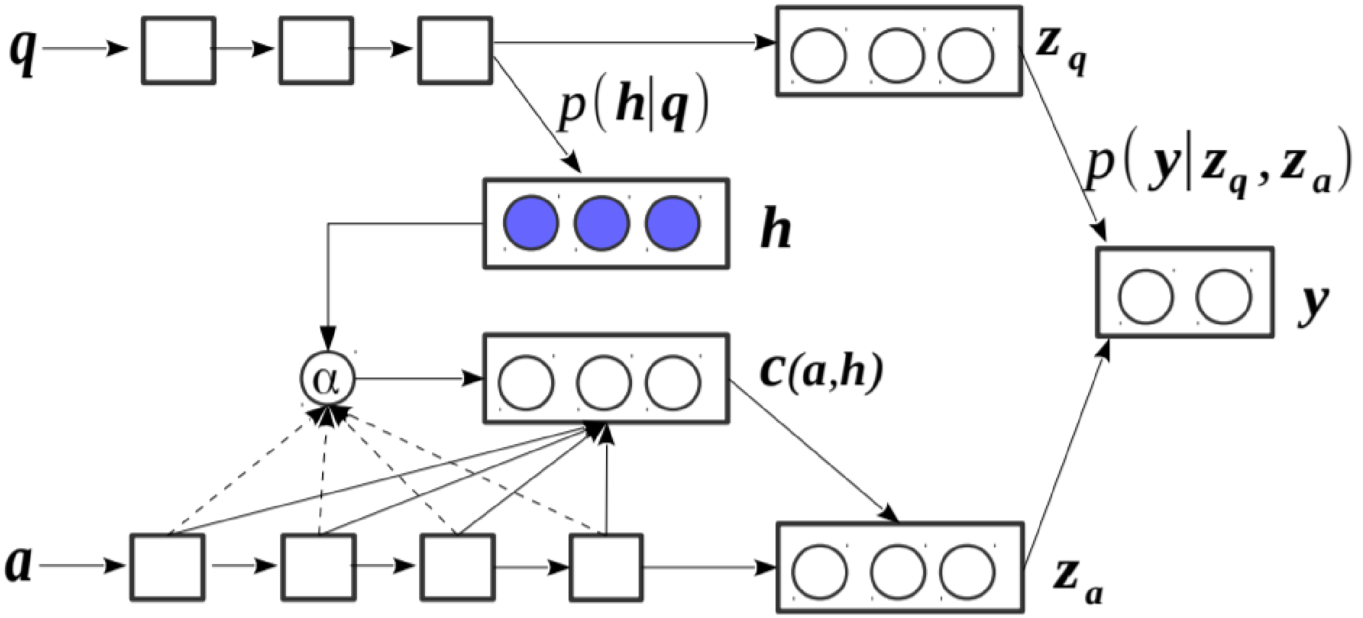


Figure 2. NASM for question answer selection.

Scenario:
Given a question q, a set of candidate answer $(a_1, a_2, \ldots, a_n)$ and judgment $(y_1, y_2, \ldots, y_n)$ where $y_m = 1$ if a_m is the answer. so each train data point is the triple(p, a, y)

Neural Variational Inference for Text Processing(2)

Model specification :

Prior：

$$p_\theta(\boldsymbol{h}|\boldsymbol{q})=\mathcal{N}(\boldsymbol{h}|\boldsymbol{\mu}(\boldsymbol{q}),\mathrm{diag}(\boldsymbol{\sigma}^2(\boldsymbol{q}))$$

$$\boldsymbol{\pi}_\theta = g_\theta(f_q^{\mathrm{LSTM}}(\boldsymbol{q})) = g_\theta(\boldsymbol{s}_q(|\boldsymbol{q}|))$$

$$\boldsymbol{\mu}_\theta = l_1(\boldsymbol{\pi}_\theta), \log\boldsymbol{\sigma}_\theta = l_2(\boldsymbol{\pi}_\theta)$$

Variational posterior：

$$q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y}) = \mathcal{N}(\boldsymbol{h}|\boldsymbol{\mu}_\phi(\boldsymbol{q},\boldsymbol{a},\boldsymbol{y}),\mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{q},\boldsymbol{a},\boldsymbol{y})))$$

$$\begin{aligned}\boldsymbol{\pi}_\phi &= g_\phi(f_q^{\mathrm{LSTM}}(\boldsymbol{q}), f_a^{\mathrm{LSTM}}(\boldsymbol{a}), f_y(\boldsymbol{y}))\\ &= g_\phi(\boldsymbol{s}_q(|\boldsymbol{q}|),\boldsymbol{s}_a(|\boldsymbol{a}|),\boldsymbol{s}_y)\\ \boldsymbol{\mu}_\phi &= l_3(\boldsymbol{\pi}_\phi), \log\boldsymbol{\sigma}_\phi = l_4(\boldsymbol{\pi}_\phi)\end{aligned}$$

Generative process:

$$\alpha(i) \propto \exp(\boldsymbol{W}_\alpha^T \tanh(\boldsymbol{W}_h\boldsymbol{h} + \boldsymbol{W}_s\boldsymbol{s}_a(i)))$$

$$\boldsymbol{c}(\boldsymbol{a},\boldsymbol{h}) = \sum\nolimits_i \boldsymbol{s}_a(i)\alpha(i)$$

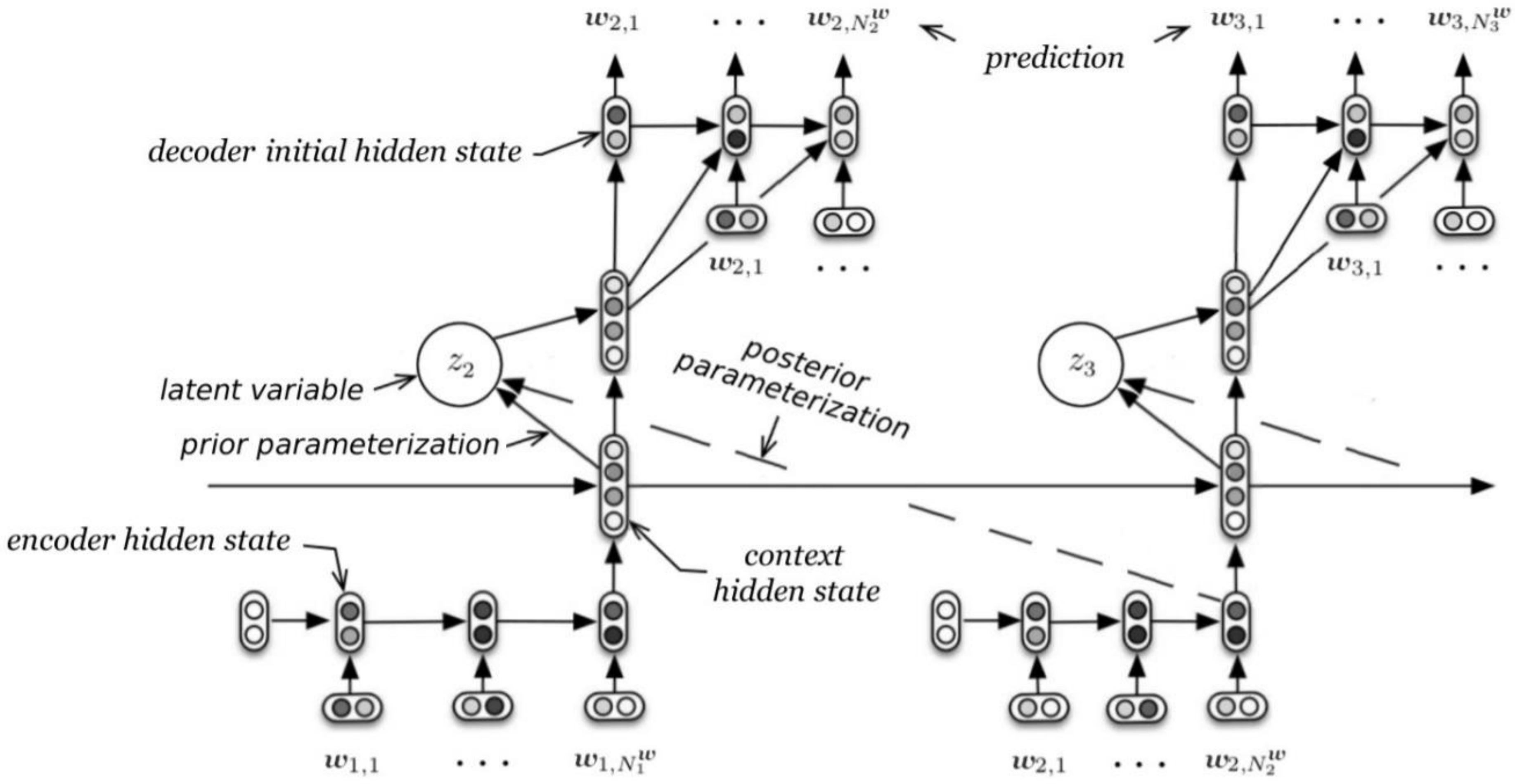
$$\boldsymbol{z}_a(\boldsymbol{a},\boldsymbol{h}) = \tanh(\boldsymbol{W}_a\boldsymbol{c}(\boldsymbol{a},\boldsymbol{h}) + \boldsymbol{W}_n\boldsymbol{s}_a(|\boldsymbol{a}|))$$

$$p_\theta(\boldsymbol{y}=1|\boldsymbol{z}_q,\boldsymbol{z}_a) = \sigma\left(\boldsymbol{z}_q^T\mathbf{M}\boldsymbol{z}_a + b\right)$$

ELBO：

$$\mathcal{L}=\mathbb{E}_{q_\phi(\boldsymbol{h})}[\log p_\theta(\boldsymbol{y}|\boldsymbol{z}_q(\boldsymbol{q}),\boldsymbol{z}_a(\boldsymbol{a},\boldsymbol{h}))]-D_{\mathrm{KL}}(q_\phi(\boldsymbol{h})||p_\theta(\boldsymbol{h}|\boldsymbol{q}))$$

Neural Variational Inference for generating dialogues



AAAI 2017

Neural Variational Inference for generating dialogues

Prior：
$$P_\theta(\mathbf{z}_n \mid \mathbf{w}_1, \ldots, \mathbf{w}_{n-1}) = \mathcal{N}(\boldsymbol{\mu}_{\text{prior}}(\mathbf{w}_1, \ldots, \mathbf{w}_{n-1}), \Sigma_{\text{prior}}(\mathbf{w}_1, \ldots, \mathbf{w}_{n-1})),$$

Generative process:
$$P_\theta(\mathbf{w}_n \mid \mathbf{z}_n, \mathbf{w}_1, \ldots, \mathbf{w}_{n-1}) = \prod_{m=1}^{M_n} P_\theta(w_{n,m} \mid \mathbf{z}_n, \mathbf{w}_1, \ldots, \mathbf{w}_{n-1}, w_{n,1}, \ldots, w_{n,m-1})$$

Variational posterior：
$$Q_\psi(\mathbf{z}_n \mid \mathbf{w}_1, \ldots, \mathbf{w}_n) = \mathcal{N}(\boldsymbol{\mu}_{\text{posterior}}(\mathbf{w}_1, \ldots, \mathbf{w}_n), \Sigma_{\text{posterior}}(\mathbf{w}_1, \ldots, \mathbf{w}_n))$$

ELBO：
$$\log P_\theta(\mathbf{w}_1, \ldots, \mathbf{w}_N) \geq \sum_{n=1}^{N} -\mathrm{KL}\left[Q_\psi(\mathbf{z}_n \mid \mathbf{w}_1, \ldots, \mathbf{w}_n) || P_\theta(\mathbf{z}_n \mid \mathbf{w}_1, \ldots, \mathbf{w}_{n-1})\right] + \mathbb{E}_{Q_\psi(\mathbf{z}_n \mid \mathbf{w}_1, \ldots, \mathbf{w}_n)}\left[\log P_\theta(\mathbf{w}_n \mid \mathbf{z}_n, \mathbf{w}_1, \ldots, \mathbf{w}_{n-1})\right],$$

Neural Variational Topic Model

Document topic distribution is a multinomial(discrete),so just transform the Gaussian variable by softmax, the rest is the same.

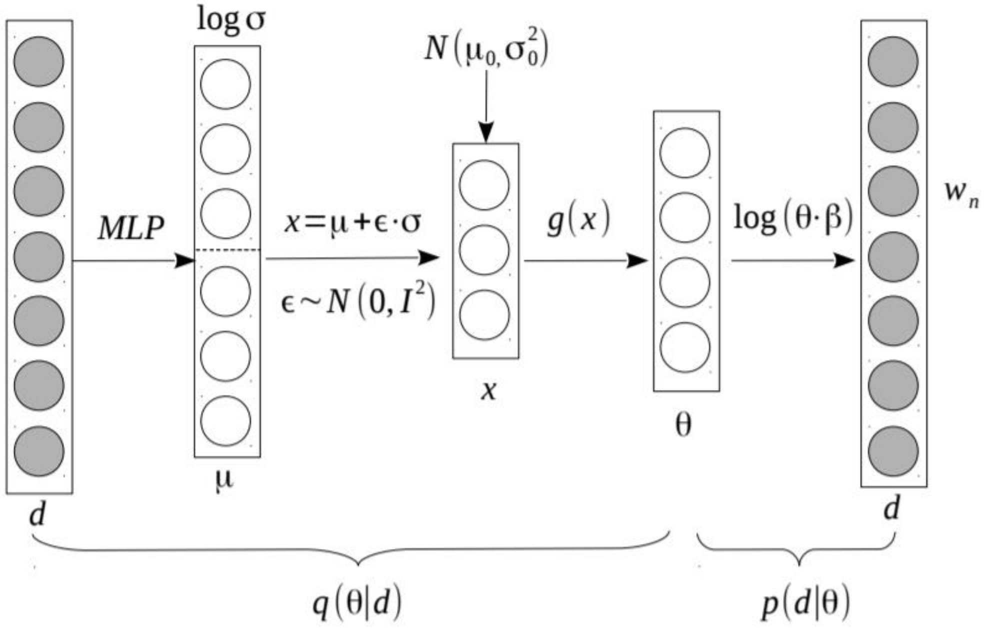


Figure 3. Network structure of the inference model $q(\theta \mid d)$, and of the generative model $p(d \mid \theta)$.

$g(x)$ is the transform function:

$$x \sim \mathcal{N}(\mu_0, \sigma_0^2)$$
$$\theta = \text{softmax}(W_1^T x)$$

ELBO:

$$\mathcal{L}_d = \mathbb{E}_{q(\theta|d)} \left[\sum_{n=1}^{N} \log \sum_{z_n} [p(w_n|\beta_{z_n}) p(z_n|\theta)] \right] - D_{KL} \left[q(\theta|d) || p(\theta|\mu_0, \sigma_0^2) \right]$$

ICML 2017
ICLR 2018 UnderReview

Neural Variational Topic Model(non-parametric version)

- Stick Breaking Process

$$\nu_k \sim \mathrm{Beta}(1, \alpha) \qquad \pi_k = \nu_k \prod_{l=1}^{k-1}(1 - \nu_l) = \nu_k(1 - \sum_{l=1}^{k-1} \pi_l)$$

$$\boldsymbol{\pi} = \{\pi_k\}_{k=1}^{\infty} \qquad 0 \leq \pi_k \leq 1 \text{ and } \sum_{k=1}^{\infty} \pi_k = 1$$

- Kumaraswamy distribution(similar to Beta distribution,more suitable for reparameterization trick)

$$\mathrm{Kumaraswamy}(x; a, b) = abx^{a-1}(1 - x^a)^{b-1}$$

Inverse CDF: $x = (1 - u^{\frac{1}{b}})^{\frac{1}{a}}$, where $u \sim \mathrm{Uniform}(0, 1)$

Neural Variational Topic Model(non-parametric version)

Generative Story:

Stick breaking process

- Draw a topic distribution $\boldsymbol{\pi} \sim \text{GEM}(\alpha)$
- Then we get a distribution $G(\boldsymbol{\theta}; \boldsymbol{\pi}, \Theta) = \sum_{k=1}^{\infty} \pi_k \delta_{\boldsymbol{\theta}_k}(\boldsymbol{\theta})$
- For each word w_i in the document: 1) draw a topic $\hat{\boldsymbol{\theta}}_i \sim G(\boldsymbol{\theta}; \boldsymbol{\pi}, \Theta)$; 2) $w_i \sim \text{Cat}(\hat{\boldsymbol{\theta}}_i)$

Here, we want to approximate the posterior distribution of $\nu_k \sim \text{Beta}(1, \alpha)$

Prior: $p(\boldsymbol{\nu}|\alpha)$ is products of $K-1$ $\mathrm{Beta}(1,\alpha)$

Variational posterior:

$$\begin{aligned}[a_1,\ldots,a_{K-1};b_1,\ldots,b_{K-1}] &= g(\mathbf{w}_{1:N};\psi)\\ q_\psi(\boldsymbol{\nu}|\mathbf{w}_{1:N}) &= \prod_{k=1}^{K-1}\kappa(\nu_k;a_k,b_k)\end{aligned}$$

Likelihood:

$$\boldsymbol{\pi} = (\pi_1,\pi_2,\ldots,\pi_{K-1},\pi_K) = \left(\nu_1,\nu_2(1-\nu_1),\ldots,\nu_{k-1}\prod_{l=1}^{K-2}(1-\nu_l),\prod_{l=1}^{K-1}(1-\nu_l)\right)$$

$$p(\mathbf{w}_{1:N},\boldsymbol{\pi},\hat{\boldsymbol{\theta}}_{1:N}|\alpha,\Theta) = p(\boldsymbol{\pi}|\alpha)\prod_{i=1}^{N}p(w_i|\hat{\boldsymbol{\theta}}_i)p(\hat{\boldsymbol{\theta}}_i|\boldsymbol{\pi},\Theta)$$

$$p(\mathbf{w}_{1:N},\boldsymbol{\pi}|\alpha,\Theta) = p(\boldsymbol{\pi}|\alpha)\prod_{i=1}^{N}p(w_i|\boldsymbol{\pi},\Theta)$$

ELBO: $\mathcal{L}(\mathbf{w}_{1:N}|\Phi,\psi) = \mathbb{E}_{q_\psi(\boldsymbol{\nu}|\mathbf{w}_{1:N})}\left[\log p(\mathbf{w}_{1:N}|\boldsymbol{\pi},\Phi)\right] - \mathrm{KL}\left(q_\psi(\boldsymbol{\nu}|\mathbf{w}_{1:N})||p(\boldsymbol{\nu}|\alpha)\right)$

References:

1. **Auto-Encoding Variational Bayes,** ICLR 2014
2. **Generating Sentences From a Continuous Spaces**, ICLR 2016
3. **Neural Variational Inference for Text Processing**, ICML 2016
4. **A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues**, AAAI 2017
5. **Discovering Discrete Latent Topics with Neural Variational Inference**, ICML 2017
6. **A Bayesian Nonparametric Topic Model with Variational Auto-Encoders**, ICLR 2018 under review
7. **A Conditional Variational Framework for Dialog Generation**, ACL 2017