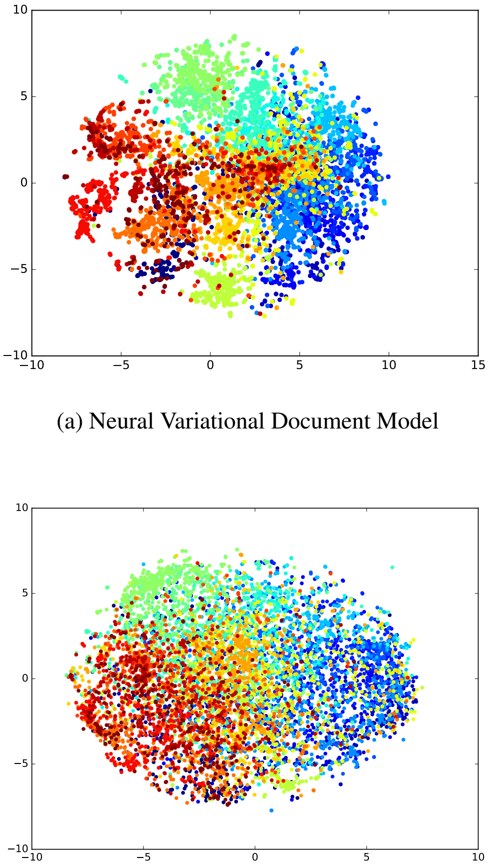
Neural Variational Inference for Text Processing (Supplementary)

Yishu Miao[1] YISHU.MIAO@CS.OX.AC.UK
Lei Yu[1] LEI.YU@CS.OX.AC.UK
Phil Blunsom[12] PHIL.BLUNSOM@CS.OX.AC.UK
[1]University of Oxford, [2]Google Deepmind

1. t-SNE Visualisation of Document Representations



(a) Neural Variational Document Model

(b) Semantic Word Vector

Figure 1. t-SNE visualisation of the document representations achieved by (a) NVDM and (b) SWV (**?**) on the held-out test dataset of *20NewsGroups*. The documents are collected from 20 different news groups, which correspond to the points with different colour in the figure.

2. Details of the Deep Neural Network Structures

2.1. Neural Variational Document Model

(1) Inference Network $q_\phi(\boldsymbol{h}|\boldsymbol{X})$:

$$\boldsymbol{\lambda} = \mathrm{ReLU}(\boldsymbol{W}_1\boldsymbol{X} + \boldsymbol{b}_1) \tag{1}$$

$$\boldsymbol{\pi} = \mathrm{ReLU}(\boldsymbol{W}_2\boldsymbol{\lambda} + \boldsymbol{b}_2) \tag{2}$$

$$\boldsymbol{\mu} = \boldsymbol{W}_3\boldsymbol{\pi} + \boldsymbol{b}_3 \tag{3}$$

$$\log\boldsymbol{\sigma} = \boldsymbol{W}_4\boldsymbol{\pi} + \boldsymbol{b}_4 \tag{4}$$

$$\boldsymbol{h} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{X}), \mathrm{diag}(\boldsymbol{\sigma}^2(\boldsymbol{X}))) \tag{5}$$

(2) Generative Model $p_\theta(\boldsymbol{X}|\boldsymbol{h})$:

$$\boldsymbol{e}_i = \exp(-\boldsymbol{h}^T\boldsymbol{R}\boldsymbol{x}_i + \boldsymbol{b}_{x_i}) \tag{6}$$

$$p_\theta(\boldsymbol{x}_i|\boldsymbol{h}) = \frac{\boldsymbol{e}_i}{\sum_j^{|V|}\boldsymbol{e}_j} \tag{7}$$

$$p_\theta(\boldsymbol{X}|\boldsymbol{h}) = \prod_i^N p_\theta(\boldsymbol{x}_i|\boldsymbol{h}) \tag{8}$$

(3) KL Divergence $D_{\mathrm{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{X})||p(\boldsymbol{h})]$:

$$D_{\mathrm{KL}} = -\tfrac{1}{2}(K - \|\boldsymbol{\mu}\|^2 - \|\boldsymbol{\sigma}\|^2 + \log|\mathrm{diag}(\boldsymbol{\sigma}^2)|) \tag{9}$$

The variational lower bound to be optimised:

$$\begin{aligned}\mathcal{L} =& \mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{X})}\Big[\sum\nolimits_{i=1}^N \log p_\theta(\boldsymbol{x}_i|\boldsymbol{h})\Big] \\ &- D_{\mathrm{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{X})||p(\boldsymbol{h})] \qquad (10)\\ \approx& \sum\nolimits_{l=1}^L \sum\nolimits_{i=1}^N \log p_\theta(\boldsymbol{x}_i|\boldsymbol{h}^{(l)}) \\ &+ \frac{1}{2}(K - \|\boldsymbol{\mu}\|^2 - \|\boldsymbol{\sigma}\|^2 + \log|\mathrm{diag}(\boldsymbol{\sigma}^2)|) \qquad (11)\end{aligned}$$

2.2. Neural Answer Selection Model

(1) Inference Network $q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y})$:

$$\boldsymbol{s}_q(|\boldsymbol{q}|) = f_q^{\text{LSTM}}(\boldsymbol{q}) \tag{12}$$

$$\boldsymbol{s}_a(|\boldsymbol{a}|) = f_a^{\text{LSTM}}(\boldsymbol{a}) \tag{13}$$

$$\boldsymbol{s}_y = \boldsymbol{W}_5\boldsymbol{y} + \boldsymbol{b}_5 \tag{14}$$

$$\boldsymbol{\gamma} = \boldsymbol{s}_q(|\boldsymbol{q}|)||\boldsymbol{s}_a(|\boldsymbol{a}|)||\boldsymbol{s}_y \tag{15}$$

$$\boldsymbol{\lambda}_\phi = \tanh(\boldsymbol{W}_6\boldsymbol{\gamma} + \boldsymbol{b}_6) \tag{16}$$

$$\boldsymbol{\pi}_\phi = \tanh(\boldsymbol{W}_7\boldsymbol{\lambda}_{\boldsymbol{\phi}} + \boldsymbol{b}_7) \tag{17}$$

$$\boldsymbol{\mu}_\phi = \boldsymbol{W}_8\boldsymbol{\pi}_{\boldsymbol{\phi}} + \boldsymbol{b}_8 \tag{18}$$

$$\log \boldsymbol{\sigma}_\phi = \boldsymbol{W}_9\boldsymbol{\pi}_{\boldsymbol{\phi}} + \boldsymbol{b}_9 \tag{19}$$

$$\boldsymbol{h} \sim \mathcal{N}(\boldsymbol{\mu}_\phi(\boldsymbol{q},\boldsymbol{a},\boldsymbol{y}), \operatorname{diag}(\boldsymbol{\sigma}^2_\phi(\boldsymbol{q},\boldsymbol{a},\boldsymbol{y}))) \tag{20}$$

(2) Generative Model

$p_\theta(\boldsymbol{h}|\boldsymbol{q})$:

$$\boldsymbol{\lambda}_\theta = \tanh(\boldsymbol{W}_1\boldsymbol{s}_q(|\boldsymbol{q}|) + \boldsymbol{b}_1) \tag{21}$$

$$\boldsymbol{\pi}_\theta = \tanh(\boldsymbol{W}_2\boldsymbol{\lambda}_\theta + \boldsymbol{b}_2) \tag{22}$$

$$\boldsymbol{\mu}_\theta = \boldsymbol{W}_3\boldsymbol{\pi}_\theta + \boldsymbol{b}_3 \tag{23}$$

$$\log \boldsymbol{\sigma}_\theta = \boldsymbol{W}_4\boldsymbol{\pi}_\theta + \boldsymbol{b}_4 \tag{24}$$

$p_\theta(\boldsymbol{y}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{h})$:

$$\boldsymbol{e}(i) = \boldsymbol{W}_\alpha^T \tanh(\boldsymbol{W}_h\boldsymbol{h} + \boldsymbol{W}_s\boldsymbol{s}_a(i)) \tag{25}$$

$$\alpha(i) = \frac{\boldsymbol{e}(i)}{\sum_j \boldsymbol{e}(j)} \tag{26}$$

$$\boldsymbol{c}(\boldsymbol{a},\boldsymbol{h}) = \textstyle\sum_i \boldsymbol{s}_a(i)\alpha(i) \tag{27}$$

$$\boldsymbol{z}_a(\boldsymbol{a},\boldsymbol{h}) = \tanh(\boldsymbol{W}_a\boldsymbol{c}(\boldsymbol{a},\boldsymbol{h}) + \boldsymbol{W}_n\boldsymbol{s}_a(|\boldsymbol{a}|)) \tag{28}$$

$$\boldsymbol{z}_q(\boldsymbol{q}) = \boldsymbol{s}_q(|\boldsymbol{q}|) \tag{29}$$

$$p_\theta(\boldsymbol{y}=1|\boldsymbol{q},\boldsymbol{a},\boldsymbol{h}) = \sigma(\boldsymbol{z}_q^T\boldsymbol{M}\boldsymbol{z}_a + b) \tag{30}$$

(3) KL Divergence $D_{\text{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y})||p_\theta(\boldsymbol{h}|\boldsymbol{q})]$:

$$\begin{aligned} D_{\text{KL}} = -\frac{1}{2}(&K + \log|\operatorname{diag}(\boldsymbol{\sigma}^2_\phi)| - \log|\operatorname{diag}(\boldsymbol{\sigma}^2_\theta)| \\ &- \operatorname{Tr}(\operatorname{diag}(\boldsymbol{\sigma}^2_\phi)\operatorname{diag}^{-1}(\boldsymbol{\sigma}^2_\theta)) \\ &- (\boldsymbol{\mu}_\phi - \boldsymbol{\mu}_\theta)^T\operatorname{diag}^{-1}(\boldsymbol{\sigma}^2_\theta)(\boldsymbol{\mu}_\phi - \boldsymbol{\mu}_\theta)) \end{aligned} \tag{31}$$

The variational lower bound to be optimised:

$$\begin{aligned} \mathcal{L} = &\mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y})}[\log p_\theta(\boldsymbol{y}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{h})] \\ &- D_{\text{KL}}[q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y})||p_\theta(\boldsymbol{h}|\boldsymbol{q})] \end{aligned} \tag{32}$$

$$\begin{aligned} \approx \sum_{l=1}^{L} &[\boldsymbol{y}\log\sigma(\boldsymbol{z}_q^T\boldsymbol{M}\boldsymbol{z}_a^{(l)} + b) \\ &+ (1-\boldsymbol{y})\log(1-\sigma(\boldsymbol{z}_q^T\boldsymbol{M}\boldsymbol{z}_a^{(l)} + b))] \\ &+ \frac{1}{2}(K + \log|\operatorname{diag}(\boldsymbol{\sigma}^2_\phi)| - \log|\operatorname{diag}(\boldsymbol{\sigma}^2_\theta)| \\ &- \operatorname{Tr}(\operatorname{diag}(\boldsymbol{\sigma}^2_\phi)\operatorname{diag}^{-1}(\boldsymbol{\sigma}^2_\theta)) \\ &- (\boldsymbol{\mu}_\phi - \boldsymbol{\mu}_\theta)^T\operatorname{diag}^{-1}(\boldsymbol{\sigma}^2_\theta)(\boldsymbol{\mu}_\phi - \boldsymbol{\mu}_\theta)) \end{aligned} \tag{33}$$

3. Computational Complexity

The computational complexity of NVDM for a training document is $C_\phi + C_\theta = O(LK^2 + KSV)$. Here, $C_\phi = O(LK^2)$ represents the cost for the inference network to generate a sample, where L is the number of the layers in the inference network and K is the average dimension of these layers. Besides, $C_\theta = O(KSV)$ is the cost of reconstructing the document from a sample, where S is the average length of the documents and V represents the volume of words applied in this document model, which is conventionally much lager than K.

The computational complexity of NASM for a training question-answer pair is $C_\phi + C_\theta = O((L+S)K^2 + SW)$. The inference network needs $C_\phi = 2SW + 2K + LK^2 = O(LK^2 + SW)$. It takes $2SW + 2K$ to produce the joint representation for a question-answer pair and its label, where W is the total number of parameters of an LSTM and S is the average length of the sentences. Based on the joint representation, an MLP spends LK^2 to generate a sample, where L is the number of layers and K represents the average dimension. The generative model requires $C_\theta = 2SW + LK^2 + SK^2 + 5K^2 + 2K^2 = O((L+S)K^2 + SW)$. Similarly, it costs $2SW + LK^2$ to construct the generative latent distribution , where $2SW$ can be saved if the LSTMs are shared by the inference network and the generative model. Besides, the attention model takes $SK^2 + 5K^2$ and the relatedness prediction takes the last $2K^2$.

Since the computations of NVDM and NASM can be parallelised in GPU and only one sample is required during training process, it is very efficient to carry out the neural variational inference. As NVDM is an instantiation of variational auto-encoder, its computational complexity is the same as the deterministic auto-encoder. In addition, the computational complexity of LSTM+Att, the deterministic counterpart of NASM, is also $O((L+S)K^2 + SW)$. There is only $O(LK^2)$ time increase by introducing an inference network for NASM when compared to LSTM+Att.