Supplementary Material: On the challenges of learning with inference networks on sparse, high-dimensional data

Contents

1 Spectral analysis of the Jacobian matrix

For any vector valued function $f(x) : \mathbb{R}^K \to \mathbb{R}^V$, $\nabla_x f(x)$ is the matrix-valued function representing the sensitivity of the output to the input. When $f(x)$ is a deep neural network, Wang *et al.* (2016) use the spectra of the Jacobian matrix under various inputs x to quantify the complexity of the learned function. They find that the spectra are correlated with the complexity of the learned function.

We adopt their technique for studying the utilization of the latent space in deep generative models. In the case of NFA, we seek to quantify the learned complexity of the generative model. To do so, we compute the Jacobian matrix as $\mathcal{J}(z) = \nabla_z \log p(x|z)$. This is a read-out measure of the sensitivity of the likelihood with respect to the latent dimension.

$\mathcal{J}(z)$ is a matrix valued function that can be evaluated at every point in the latent space. We evaluate it it at the mode of the (unimodal) prior distribution i.e. at $z = \vec{0}$. The singular values of the resulting matrix denote how much the log-likelihood changes from the origin along the singular vectors lying in latent space. The intensity of these singular values (which we plot) is a read-out measure of how many intrinsic dimensions are utilized by the model parameters θ at the mode of the prior distribution. Our choice of evaluating $\mathcal{J}(z)$ at $z = \vec{0}$ is motivated by the fact that much of the probability mass in latent space under the NFA model will be placed at the origin. We use the utilization at the mode as an approximation for the utilization across the entire latent space. We also plotted the spectral decomposition obtained under a Monte-Carlo approximation to the matrix $\mathbb{E}[\mathcal{J}(z)]$ and found it to be similar to the decomposition obtained by evaluating the Jacobian at the mode.

Another possibility to measure utilization would be from the KL divergence of the prior and the output of the inference network (as in Burda *et al.* (2015)).

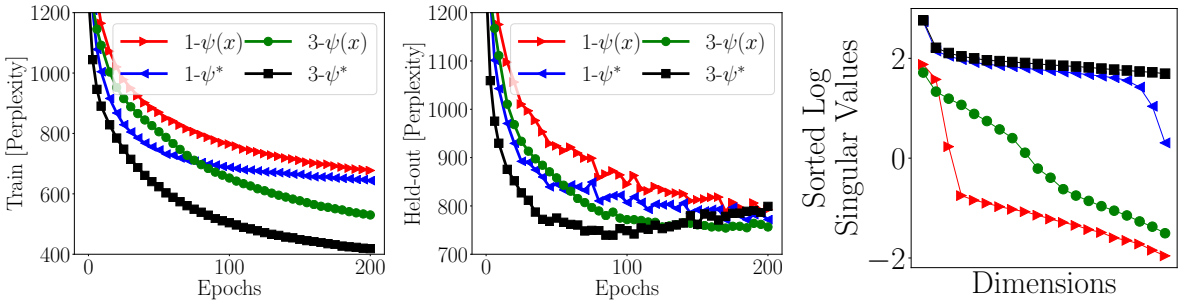
2 Learning with ψ^* on a small dataset

Table 3: **Test Perplexity on 20newsgroups: Left: Baselines** Legend: LDA (Blei *et al.* , 2003), Replicated Softmax (RSM) (Hinton & Salakhutdinov, 2009), Sigmoid Belief Networks (SBN) and Deep Autoregressive Networks (DARN) (Mnih & Gregor, 2014), Neural Variational Document Model (NVDM) (Miao *et al.* , 2016). K denotes the latent dimension in our notation. **Right:** NFA on text data with $K = 100$. We vary the features presented to the inference network $q_\phi(z|x)$ during learning between: normalized count vectors ($\frac{x}{\sum_{i=1}^{V} x_i}$, denoted "norm") and normalized TF-IDF (denoted "tfidf") features.

Model	K	**Results**
LDA	50	1091
LDA	200	1058
RSM	50	953
SBN	50	909
fDARN	50	917
fDARN	200	—
NVDM	50	836
NVDM	200	852

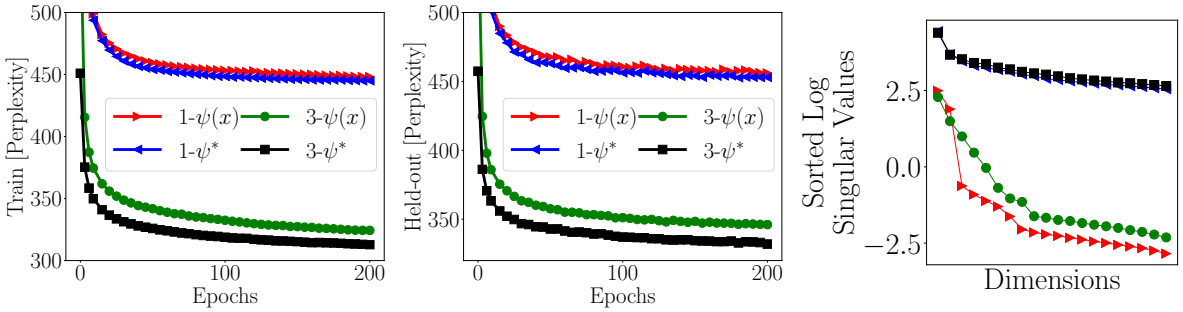
NFA	**Perplexity**	
	$\psi(x)$	ψ^*
1-$\psi(x)$-norm	1018	903
1-ψ^*-norm	1279	889
3-$\psi(x)$-norm	986	857
3-ψ^*-norm	1292	879
1-$\psi(x)$-tfidf	932	839
1-ψ^*-tfidf	953	828
3-$\psi(x)$-tfidf	999	842
3-ψ^*-tfidf	1063	839

In the main paper, we studied the optimization of variational parameters on the larger RCV1 and Wikipedia datasets. Here, we study the role of learning with ψ^* in the small-data regime. Table 3 depicts the results obtained after training models for 200 passes through the data. We summarize our findings: (1) across the board, TF-IDF features improve learning, and (2) in the small data regime, deeper non-linear models (3-ψ^*-tfidf) overfit quickly and better results are obtained by the simpler multinomial-logistic PCA model (1-ψ^*-tfidf). Overfitting is also evident in Fig. 7 from comparing curves on the validation set to those on the training set. Interestingly, in the small dataset setting, we see that learning with $\psi(x)$ has the potential to have a regularization effect in that the results obtained are not much worse than those obtained from learning with ψ^*.



(a) Training Data (b) Held-out Data (c) Log-singular Values

Figure 7: **20Newsgroups - Training and Held-out Bounds:** Fig. 7a, 7b denotes the train (held-out) perplexity for different models. Fig. 7c depicts the log-singular values of the Jacobian matrix for the trained models.



(a) Training Data (b) Held-out Data (c) Log-singular Values

Figure 8: **RCV1 - Training and Held-out Bounds:** Fig. 8a, 8b denotes the train (held-out) perplexity for different models. Fig. 8c depicts the log-singular values of the Jacobian matrix for the trained models.

For completeness, in Fig. 8, we also provide the training behavior for the RCV1 dataset corresponding to the results of Table 1 (in the main paper). The results here, echo the convergence behavior on the Wikipedia dataset.

3 Comparison with KL-annealing

An empirical observation made in previous work is that when $p(x|z;\theta)$ is complex (parameterized by a recurrent neural network or a neural autoregressive density estimator (NADE)), the generative model also must contend with overpruning of the latent dimension. A proposed fix is the annealing of the KL divergence term in Equation (2) (e.g., Bowman *et al.* , 2016) as one way to overcome local minima. As discussed in the main paper, this is a different failure mode to the one we present in that our decoder is a vanilla MLP – nonetheless, we apply KL annealing within our setting.

In particular, we optimized $\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z))\right] - \eta\,\mathrm{KL}(\,q_\phi(z|x)||p(z)\,)$ where η was annealed from 0 to 1 (linearly – though we also tried exponential annealing) over the course of several parameter updates. Note that doing so does *not* give us a lower bound on the likelihood of the data during learning until $\eta = 1$. There are few established guidelines about the rate of annealing the KL divergence and in general, we found it tricky to get it to work reliably. We experimented with different rates of annealing for learning a three-layer generative model on the Wikipedia data.

Our findings (visualized in Fig. 10) are as follows: (1) on sparse data we found annealing the KL divergence is very sensitive to the annealing rate – too small an annealing rate and we were still left with underfitting (as in annealing for 10k), too high an annealing rate (as in 100k) and this resulted in slow convergence; (2) learning with ψ^* always outperformed (in both rate of convergence and quality of final result on train and held-out data) annealing the KL divergence across various choices of annealing schedules. Said differently, on the Wikipedia dataset, we conjecture there exists a choice of annealing of the KL divergence for which the perplexity obtained *may* match those of learning with ψ^* but finding this schedule requires significant trial and error – Fig. 10 suggests that we did not find it. We found that learning with ψ^* was more robust, required less tuning (setting values of M to be larger than 100 never hurt) and always performed at par or better than annealing the KL divergence. Furthermore, we did not find annealing the KL divergence to 1 to relize

good results for the experiments on the recommender systems task.

4 Depth of $q_\phi(z|x)$

Can the overall effect of the additional optimization be *learned* by the inference network at training time? The experimental evidence we observe in Fig. 9 suggests this is difficult.

When learning with $\psi(x)$, increasing the number of layers in the inference network slightly decreases the quality of the model learned. This is likely because the already stochastic gradients of the inference network must propagate along a longer path in a deeper inference network, slowing down learning of the parameters ϕ which in turn affects $\psi(x)$, thereby reducing the quality of the gradients used to updated θ.

5 Inference on documents with rare words

We sample 20000 training and held-out data points; we compute $\mathrm{KL}(\psi(x)\|\psi^*)$ (both are Normal distributions and the KL is analytic) and the number of rare words in each document (where a word is classified as being rare if it occurs in less than 5% of training documents). For each document, we normalize both values to be between 0 and 1 using: $\frac{c_i - min(c)}{max(c) - min(c)}$ where c is the vector of KL divergences or number of rare words. We sort the paired values by the KL divergence and plot them in Fig. 11. The x-axis corresponds to the index of the sampled datapoints and the y-axis corresponds to the normalized value of both the KL divergence and the number of rare words in the document. As before, we observe that the documents that the inference network's initial parameters are worst on (for which the variational parameters must change the most –measured in KL divergence) are those which have many rare words.

References

Blei, David M, Ng, Andrew Y, & Jordan, Michael I. 2003. Latent dirichlet allocation. *JMLR*.

Bowman, Samuel R, Vilnis, Luke, Vinyals, Oriol, Dai, Andrew M, Jozefowicz, Rafal, & Bengio, Samy. 2016. Generating sentences from a continuous space. *In: CoNLL*.

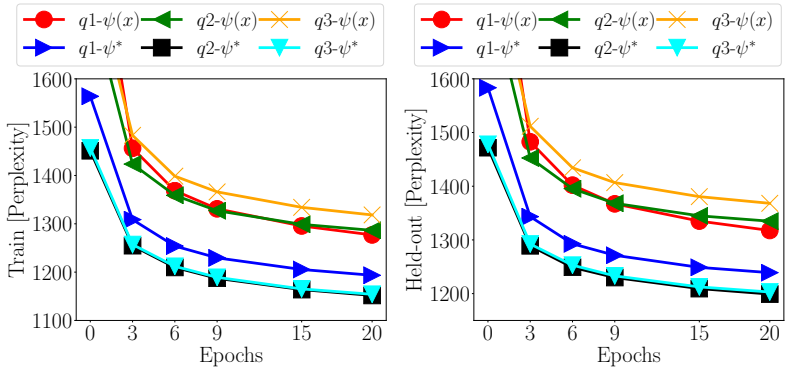
Burda, Yuri, Grosse, Roger, & Salakhutdinov, Ruslan. 2015. Importance weighted autoencoders. *In: ICLR*.

Hinton, Geoffrey E, & Salakhutdinov, Ruslan R. 2009. Replicated softmax: an undirected topic model. *In: NIPS*.

Miao, Yishu, Yu, Lei, & Blunsom, Phil. 2016. Neural Variational Inference for Text Processing. *In: ICML*.

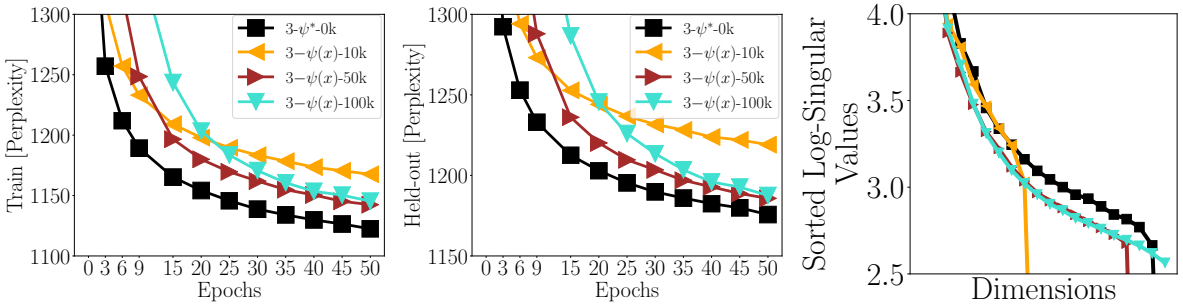
Mnih, Andriy, & Gregor, Karol. 2014. Neural variational inference and learning in belief networks. *In: ICML*.

Wang, Shengjie, Plilipose, Matthai, Richardson, Matthew, Geras, Krzysztof, Urban, Gregor, & Aslan, Ozlem. 2016. Analysis of Deep Neural Networks with the Extended Data Jacobian Matrix. *In: ICML*.



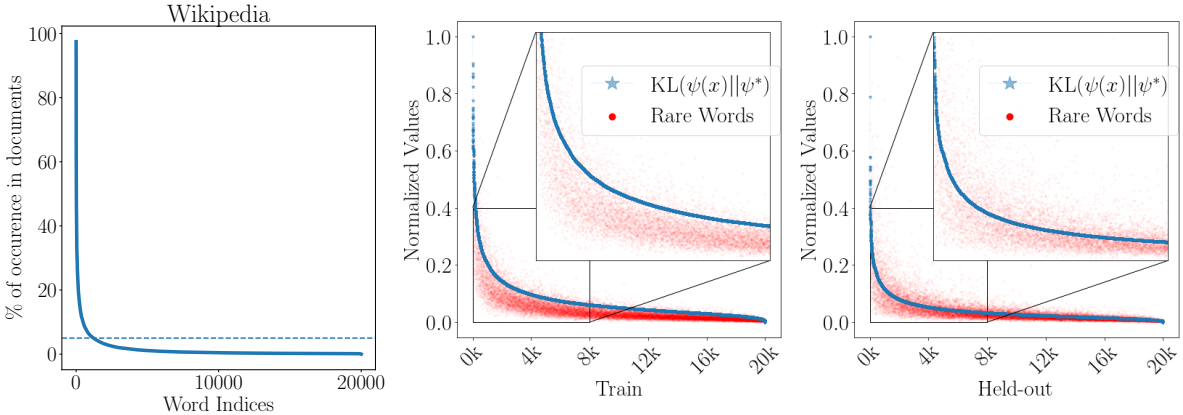
(a) Training Data (b) Held-out Data

Figure 9: **Varying the Depth of $q_\phi(z|x)$:** Fig. 10a (10b) denotes the train (held-out) perplexity for a three-layer generative model learned with inference networks of varying depth. The notation q3-ψ^* denotes that the inference network contained a two-layer intermediate hidden layer $h(x) = \text{MLP}(x; \phi_0)$ followed by $\mu(x) = W_\mu h(x), \log \Sigma(x) = W_{\log \Sigma} h(x)$.



(a) Training Data (b) Held-out Data (c) Log-singular Values

Figure 10: **KL annealing vs learning with ψ^*** Fig. 10a, 10b denotes the train (held-out) perplexity for different training methods. The suffix at the end of the model configuration denotes the number of parameter updates that it took for the KL divergence to be annealed from 0 to 1. 3-ψ^*-50k denotes that it took 50000 parameter updates before $-\mathcal{L}(x; \theta, \psi(x))$ was used as the loss function. Fig. 7c depicts the log-singular values of the Jacobian matrix for the trained models.



(a) Sparsity of Wikipedia (b) Training Data (c) Held-out Data

Figure 11: **Normalized KL and Rare Word Counts:** Fig. 11a depicts percentage of times words appear in the Wikipedia dataset (sorted by frequency). The dotted line in blue denotes the marker for 5%. In Fig. 11b, 11c, we superimpose and compare on the y-axis (1) the normalized (to be between 0 and 1) values of $\text{KL}(\psi(x)\|\psi^*)$ and (2) the normalized number of rare words (sorted by value of the KL-divergence) in a document for 20,000 points (on the x-axis) randomly sampled from the train and held-out data.