Piecewise Latent Variables for Neural Variational Text Processing

Iulian V. Serban[1*] and **Alexander G. Ororbia II**[2*] and **Joelle Pineau**[3] and **Aaron Courville**[1]
[1] Department of Computer Science and Operations Research, Universite de Montreal
[2]College of Information Sciences & Technology, Penn State University
[3]School of Computer Science, McGill University
iulian [DOT] vlad [DOT] serban [AT] umontreal [DOT] ca
ago109 [AT] psu [DOT] edu
jpineau [AT] cs [DOT] mcgill [DOT] ca
aaron [DOT] courville [AT] umontreal [DOT] ca

Abstract

Advances in neural variational inference have facilitated the learning of powerful directed graphical models with continuous latent variables, such as variational autoencoders. The hope is that such models will learn to represent rich, multi-modal latent factors in real-world data, such as natural language text. However, current models often assume simplistic priors on the latent variables — such as the uni-modal Gaussian distribution — which are incapable of representing complex latent factors efficiently. To overcome this restriction, we propose the simple, but highly flexible, piecewise constant distribution. This distribution has the capacity to represent an exponential number of modes of a latent target distribution, while remaining mathematically tractable. Our results demonstrate that incorporating this new latent distribution into different models yields substantial improvements in natural language processing tasks such as document modeling and natural language generation for dialogue.

1 Introduction

The development of the variational autoencoder framework (Kingma and Welling, 2014; Rezende et al., 2014) has paved the way for learning large-scale, directed latent variable models. This has led to significant progress in a diverse set of machine learning applications, ranging from computer vision (Gregor et al., 2015; Larsen et al., 2016) to natural language processing tasks (Mnih and Gregor, 2014; Miao et al., 2016; Bowman et al., 2015; Serban et al., 2017b). It is hoped that this framework will enable the learning of generative processes of real-world data — including text, audio and images — by disentangling and representing the underlying latent factors in the data. However, latent factors in real-world data are often highly complex. For example, topics in newswire text and responses in conversational dialogue often posses latent factors that follow non-linear (non-smooth), multi-modal distributions (i.e. distributions with multiple local maxima).

Nevertheless, the majority of current models assume a simple prior in the form of a multivariate Gaussian distribution in order to maintain mathematical and computational tractability. This is often a highly restrictive and unrealistic assumption to impose on the structure of the latent variables. First, it imposes a strong uni-modal structure on the latent variable space; latent variable samples from the generating model (prior distribution) all cluster around a single mean. Second, it forces the latent variables to follow a perfectly symmetric distribution with constant kurtosis; this makes it difficult to represent asymmetric or rarely occurring factors. Such constraints on the latent variables increase pressure on the down-stream generative model, which in turn is forced to carefully partition the probability mass for each latent factor throughout its intermediate layers. For complex, multi-modal distributions — such as the distribution over topics in a text corpus, or natural language responses in a dialogue system — the uni-modal Gaussian prior inhibits the model's ability to extract and represent important latent structure in the data. In order to learn more expressive latent variable models, we therefore need more flexible, yet tractable, priors.

In this paper, we introduce a simple, flexible

[*] The first two authors contributed equally.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 420–430
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

prior distribution based on the piecewise constant distribution. We derive an analytical, tractable form that is applicable to the variational autoencoder framework and propose a differentiable parametrization for it. We then evaluate the effectiveness of the distribution when utilized both as a prior and as approximate posterior across variational architectures in two natural language processing tasks: document modeling and natural language generation for dialogue. We show that the piecewise constant distribution is able to capture elements of a target distribution that cannot be captured by simpler priors — such as the uni-modal Gaussian. We demonstrate state-of-the-art results on three document modeling tasks, and show improvements on a dialogue natural language generation. Finally, we illustrate qualitatively how the piecewise constant distribution represents multi-modal latent structure in the data.

2 Related Work

The idea of using an artificial neural network to approximate an inference model dates back to the early work of Hinton and colleagues (Hinton and Zemel, 1994; Hinton et al., 1995; Dayan and Hinton, 1996). Researchers later proposed Markov chain Monte Carlo methods (MCMC) (Neal, 1992), which do not scale well and mix slowly, as well as variational approaches which require a tractable, factored distribution to approximate the true posterior distribution (Jordan et al., 1999). Others have since proposed using feed-forward inference models to initialize the mean-field inference algorithm for training Boltzmann architectures (Salakhutdinov and Larochelle, 2010; Ororbia II et al., 2015). Recently, the variational autoencoder framework (VAE) was proposed by Kingma and Welling (2014) and Rezende et al. (2014), closely related to the method proposed by Mnih and Gregor (2014). This framework allows the joint training of an inference network and a directed generative model, maximizing a variational lower-bound on the data log-likelihood and facilitating exact sampling of the variational posterior. Our work extends this framework.

With respect to document modeling, neural architectures have been shown to outperform well-established topic models such as Latent Dirichlet Allocation (LDA) (Hofmann, 1999; Blei et al., 2003). Researchers have successfully proposed several models involving discrete latent variables (Salakhutdinov and Hinton, 2009; Hinton and Salakhutdinov, 2009; Srivastava et al., 2013; Larochelle and Lauly, 2012; Uria et al., 2014; Lauly et al., 2016; Bornschein and Bengio, 2015; Mnih and Gregor, 2014). The success of such discrete latent variable models — which are able to partition probability mass into separate regions — serves as one of our main motivations for investigating models with more flexible continuous latent variables for document modeling. More recently, Miao et al. (2016) proposed to use continuous latent variables for document modeling.

Researchers have also investigated latent variable models for dialogue modeling and dialogue natural language generation (Bangalore et al., 2008; Crook et al., 2009; Zhai and Williams, 2014). The success of discrete latent variable models in this task also motivates our investigation of more flexible continuous latent variables. Closely related to our proposed approach is the Variational Hierarchical Recurrent Encoder-Decoder (*VHRED*, described below) (Serban et al., 2017b), a neural architecture with latent multivariate Gaussian variables.

Researchers have explored more flexible distributions for the latent variables in VAEs, such as autoregressive distributions, hierarchical probabilistic models and approximations based on MCMC sampling (Rezende et al., 2014; Rezende and Mohamed, 2015; Kingma et al., 2016; Ranganath et al., 2016; Maaløe et al., 2016; Salimans et al., 2015; Burda et al., 2016; Chen et al., 2017; Ruiz et al., 2016). These are all complimentary to our approach; it is possible to combine them with the piecewise constant latent variables. In parallel to our work, multiple research groups have also proposed VAEs with discrete latent variables (Maddison et al., 2017; Jang et al., 2017; Rolfe, 2017; Johnson et al., 2016). This is a promising line of research, however these approaches often require approximations which may be inaccurate when applied to larger scale tasks, such as document modeling or natural language generation. Finally, discrete latent variables may be inappropriate for certain natural language processing tasks.

3 Neural Variational Models

We start by introducing the neural variational learning framework. We focus on modeling discrete output variables (e.g. words) in the context of natural language processing applications. However, the framework can easily be adapted to handle continuous output variables.

3.1 Neural Variational Learning

Let $w_1, \ldots, w_N$ be a sequence of N tokens (words) conditioned on a continuous latent variable z. Further, let c be an additional observed variable which conditions both z and $w_1, \ldots, w_N$. Then, the distribution over words is:

$$P_\theta(w_1, \ldots, w_N|c) = \int \prod_{n=1}^{N} P_\theta(w_n|w_{<n}, z, c) P_\theta(z|c) dz,$$

where θ are the model parameters. The model first generates the higher-level, continuous latent variable z conditioned on c. Given z and c, it then generates the word sequence $w_1, \ldots, w_N$. For unsupervised modeling of documents, the c is excluded and the words are assumed to be independent of each other, when conditioned on z:

$$P_\theta(w_1, \ldots, w_N) = \int \prod_{n=1}^{N} P_\theta(w_n|z) P_\theta(z) dz.$$

Model parameters can be learned using the variational lower-bound (Kingma and Welling, 2014):

$$\begin{aligned} &\log P_\theta(w_1, \ldots, w_N|c) \\ &\quad \geq \quad \mathrm{E}_{z \sim Q_\psi(z|w_1,\ldots,w_N,c)}[\log P_\theta(w_n|w_{<n}, z, c)] \\ &\qquad - \mathrm{KL}\left[Q_\psi(z|w_1, \ldots, w_N, c)||P_\theta(z|c)\right], \qquad (1) \end{aligned}$$

where we note that $Q_\psi(z|w_1, \ldots, w_N, c)$ is the approximation to the intractable, true posterior $P_\theta(z|w_1, \ldots, w_N, c)$. Q is called the *encoder*, or sometimes the *recognition model* or *inference model*, and it is parametrized by ψ. The distribution $P_\theta(z|c)$ is the prior model for z, where the only available information is c. The VAE framework further employs the re-parametrization trick, which allows one to move the derivative of the lower-bound inside the expectation. To accomplish this, z is parametrized as a transformation of a fixed, parameter-free random distribution $z = f_\theta(\epsilon)$, where ϵ is drawn from a random distribution. Here, f is a transformation of ϵ, parametrized by θ, such that $f_\theta(\epsilon) \sim P_\theta(z|c)$. For example, ϵ might be drawn from a standard Gaussian distribution and f might be defined as $f_\theta(\epsilon) = \mu + \sigma\epsilon$, where μ and σ are in the parameter set θ. In this case, z is able to represent any Gaussian with mean μ and variance σ^2.

Model parameters are learned by maximizing the variational lower-bound in eq. (1) using gradient descent, where the expectation is computed using samples from the approximate posterior.

The majority of work on VAEs propose to parametrize z as multivariate Gaussian distributions. However, this unrealistic assumption may critically hurt the expressiveness of the latent variable model. See Appendix A for a detailed discussion. This motivates the proposed piecewise constant latent variable distribution.

3.2 Piecewise Constant Distribution

We propose to learn latent variables by parametrizing z using a piecewise constant probability density function (PDF). This should allow z to represent complex aspects of the data distribution in latent variable space, such as non-smooth regions of probability mass and multiple modes.

Let $n \in \mathbb{N}$ be the number of piecewise constant components. We assume z is drawn from PDF:

$$P(z) = \frac{1}{K} \sum_{i=1}^{n} 1_{\left(\frac{i-1}{n} \leq z \leq \frac{i}{n}\right)} a_i, \qquad (2)$$

where $1_{(x)}$ is the indicator function, which is one when x is true and otherwise zero. The distribution parameters are $a_i > 0$, for $i = 1, \ldots, n$. The normalization constant is:

$$K = \sum_{i=1}^{n} K_i, \text{ where } K_0 = 0, K_i = \frac{a_i}{n}, \text{ for } i = 1, \ldots, n.$$

It is straightforward to show that a piecewise constant distribution with more than $n > 2$ pieces is capable of representing a bi-modal distribution. When $n > 2$, a vector z of piecewise constant variables can represent a probability density with $2^{|z|}$ modes. Figure 1 illustrates how these variables help model complex, multi-modal distributions.

In order to compute the variational bound, we need to draw samples from the piecewise constant distribution using its inverse cumulative distribution function (CDF). Further, we need to compute the KL divergence between the prior and posterior. The inverse CDF and KL divergence quantities are both derived in Appendix B. During training we must compute derivatives of the variational bound in eq. (1). These expressions involve derivatives of indicator functions, which have derivatives zero everywhere except for the changing points where the derivative is undefined. However, the probability of sampling the value exactly at its changing

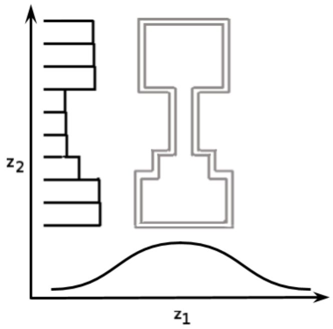

Figure 1: Joint density plot of a pair of Gaussian and piecewise constant variables. The horizontal axis corresponds to z_1, which is a univariate Gaussian variable. The vertical axis corresponds to z_2, which is a piecewise constant variable.

point is effectively zero. Thus, we fix these derivatives to zero. Similar approximations are used in training networks with rectified linear units.

4 Latent Variable Parametrizations

In this section, we develop the parametrization of both the Gaussian variable and our proposed piecewise constant latent variable.

Let x be the current output sequence, which the model must generate (e.g. $w_1, \ldots, w_N$). Let c be the observed conditioning information. If the task contains additional conditioning information this will be embedded by c. For example, for dialogue natural language generation c represents an embedding of the dialogue history, while for document modeling $c = \emptyset$.

4.1 Gaussian Parametrization

Let μ^{prior} and $\sigma^{2,\text{prior}}$ be the prior mean and variance, and let μ^{post} and $\sigma^{2,\text{post}}$ be the approximate posterior mean and variance. For Gaussian latent variables, the prior distribution mean and variances are encoded using linear transformations of a hidden state. In particular, the prior distribution covariance is encoded as a diagonal covariance matrix using a softplus function:

$$\begin{aligned}\mu^{\text{prior}} &= H_\mu^{\text{prior}}\text{Enc}(c) + b_\mu^{\text{prior}},\\ \sigma^{2,\text{prior}} &= \text{diag}(\log(1+\exp(H_\sigma^{\text{prior}}\text{Enc}(c) + b_\sigma^{\text{prior}}))),\end{aligned}$$

where $\text{Enc}(c)$ is an embedding of the conditioning information c (e.g. for dialogue natural language generation this might, for example, be produced by an LSTM encoder applied to the dialogue history), which is shared across all latent variable dimensions. The matrices $H_\mu^{\text{prior}}, H_\sigma^{\text{prior}}$ and vectors $b_\mu^{\text{prior}}, b_\sigma^{\text{prior}}$ are learnable parameters. For the posterior distribution, previous work has shown it is better to parametrize the posterior distribution as a linear interpolation of the prior distribution mean and variance and a new estimate of the mean and variance based on the observation x (Fraccaro et al., 2016). The interpolation is controlled by a gating mechanism, allowing the model to turn on/off latent dimensions:

$$\begin{aligned}\mu^{\text{post}} =& (1-\alpha_\mu)\mu^{\text{prior}} + \alpha_\mu\left(H_\mu^{\text{post}}\text{Enc}(c,x) + b_\mu^{\text{post}}\right),\\ \sigma^{2,\text{post}} =& (1-\alpha_\sigma)\sigma^{2,\text{prior}}\\ &+ \alpha_\sigma\text{diag}(\log(1+\exp(H_\sigma^{\text{post}}\text{Enc}(c,x) + b_\sigma^{\text{post}}))),\end{aligned}$$

where $\text{Enc}(c,x)$ is an embedding of both c and x. The matrices $H_\mu^{\text{post}}, H_\sigma^{\text{post}}$ and the vectors $b_\mu^{\text{post}}, b_\sigma^{\text{post}}, \alpha_\mu, \alpha_\sigma$ are parameters to be learned. The interpolation mechanism is controlled by α_μ and α_σ, which are initialized to zero (i.e. initialized such that the posterior is equal to the prior).

4.2 Piecewise Constant Parametrization

We parametrize the piecewise prior parameters using an exponential function applied to a linear transformation of the conditioning information:

$$a_i^{\text{prior}} = \exp(H_{a,i}^{\text{prior}}\text{Enc}(c) + b_{a,i}^{\text{prior}}), \quad i = 1, \ldots, n,$$

where matrix H_a^{prior} and vector b_a^{prior} are learnable. As before, we define the posterior parameters as a function of both c and x:

$$a_i^{\text{post}} = \exp(H_{a,i}^{\text{post}}\text{Enc}(c,x) + b_{a,i}^{\text{post}}), \quad i = 1, \ldots, n,$$

where H_a^{post} and b_a^{post} are parameters.

5 Variational Text Modeling

We now introduce two classes of VAEs. The models are extended by incorporating the Gaussian and piecewise latent variable parametrizations.

5.1 Document Model

The neural variational document model (*NVDM*) model has previously been proposed for document modeling (Mnih and Gregor, 2014; Miao et al., 2016), where the latent variables are Gaussian. Since the original *NVDM* uses Gaussian latent variables, we will refer to it as *G-NVDM*. We propose two novel models building on *G-NVDM*. The first model we propose uses piecewise constant latent variables instead of Gaussian latent variables.

We refer to this model as *P-NVDM*. The second model we propose uses a combination of Gaussian and piecewise constant latent variables. The models sample the Gaussian and piecewise constant latent variables independently and then concatenates them together into one vector. We refer to this model as *H-NVDM*.

Let V be the vocabulary of document words. Let W represent a document matrix, where row w_i is the 1-of-$|V|$ binary encoding of the i'th word in the document. Each model has an encoder component $Enc(W)$, which compresses a document vector into a continuous distributed representation upon which the approximate posterior is built. For document modeling, word order information is not taken into account and no additional conditioning information is available. Therefore, each model uses a bag-of-words encoder, defined as a multi-layer perceptron (MLP) $Enc(c = \emptyset, x) = Enc(x)$. Based on preliminary experiments, we choose the encoder to be a two-layered MLP with parametrized rectified linear activation functions (we omit these parameters for simplicity). For the approximate posterior, each model has the parameter matrix W_a^{post} and vector b_a^{post} for the piecewise latent variables, and the parameter matrices $W_\mu^{\text{post}}, W_\sigma^{\text{post}}$ and vectors $b_\mu^{\text{post}}, b_\sigma^{\text{post}}$ for the Gaussian means and variances. For the prior, each model has parameter vector b_a^{prior} for the piecewise latent variables, and vectors $b_\mu^{\text{prior}}, b_\sigma^{\text{prior}}$ for the Gaussian means and variances. We initialize the bias parameters to zero in order to start with centered Gaussian and piecewise constant priors. The encoder will adapt these priors as learning progresses, using the gating mechanism to turn on/off latent dimensions.

Let z be the vector of latent variables sampled according to the approximate posterior distribution. Given z, the decoder $Dec(w, z)$ outputs a distribution over words in the document:

$$Dec(w, z) = \frac{\exp\left(-w^{\text{T}} R z + b_w\right)}{\sum_{w'} \exp\left(-w^{\text{T}} R z + b_{w'}\right)},$$

where R is a parameter matrix and b is a parameter vector corresponding to the bias for each word to be learned. This output probability distribution is combined with the KL divergences to compute the lower-bound in eq. (1). See Appendix C.

Our baseline model *G-NVDM* is an improvement over the original *NVDM* proposed by Mnih and Gregor (2014) and Miao et al. (2016). We learn the prior mean and variance, while these were fixed to a standard Gaussian in previous work. This increases the flexibility of the model and makes optimization easier. In addition, we use a gating mechanism for the approximate posterior of the Gaussian variables. This gating mechanism allows the model to turn off latent variable (i.e. fix the approximate posterior to equal the prior for specific latent variables) when computing the final posterior parameters. Furthermore, Miao et al. (2016) alternated between optimizing the approximate posterior parameters and the generative model parameters, while we optimize all parameters simultaneously.

5.2 Dialogue Model

The variational hierarchical recurrent encoder-decoder (*VHRED*) model has previously been proposed for dialogue modeling and natural language generation (Serban et al., 2017b, 2016a). The model decomposes dialogues using a two-level hierarchy: sequences of utterances (e.g. sentences), and sub-sequences of tokens (e.g. words). Let $\mathbf{w}_n$ be the n'th utterance in a dialogue with N utterances. Let $w_{n,m}$ be the m'th word in the n'th utterance from vocabulary V given as a 1-of-$|V|$ binary encoding. Let M_n be the number of words in the n'th utterance. For each utterance $n = 1, \dots, N$, the model generates a latent variable z_n. Conditioned on this latent variable, the model then generates the next utterance:

$$P_\theta(\mathbf{w}_1, z_1, \dots, \mathbf{w}_N, z_N) = \prod_{n=1}^{N} P_\theta(z_n | \mathbf{w}_{<n}) \times \prod_{m=1}^{M_n} P_\theta(w_{n,m} | w_{n,<m}, \mathbf{w}_{<n}, z_n),$$

where θ are the model parameters. *VHRED* consists of three RNN modules: an *encoder* RNN, a *context* RNN and a *decoder* RNN. The *encoder* RNN computes an embedding for each utterance. This embedding is fed into the *context* RNN, which computes a hidden state summarizing the dialogue context before utterance n: h_{n-1}^{con}. This state represents the additional conditioning information, which is used to compute the prior distribution over z_n:

$$P_\theta(z_n \mid \mathbf{w}_{<n}) = f_\theta^{\text{prior}}(z_n; h_{n-1}^{con}),$$

where f^{prior} is a PDF parametrized by both θ and h_{n-1}^{con}. A sample is drawn from this distribution: $z_n \sim P_\theta(z_n | \mathbf{w}_{<n})$. This sample is given as input

to the *decoder* RNN, which then computes the output probabilities of the words in the next utterance. The model is trained by maximizing the variational lower-bound, which factorizes into independent terms for each sub-sequence (utterance):

$$\begin{aligned}&\log P_\theta(\mathbf{w}_1,\ldots,\mathbf{w}_N)\\ &\geq \sum_{n=1}^{N} - \mathrm{KL}\left[Q_\psi(z_n \mid \mathbf{w}_1,\ldots,\mathbf{w}_n) || P_\theta(z_n \mid \mathbf{w}_{<n})\right]\\ &\qquad + \mathbb{E}_{Q_\psi(z_n|\mathbf{w}_1,\ldots,\mathbf{w}_n)}\left[\log P_\theta(\mathbf{w}_n \mid z_n, \mathbf{w}_{<n})\right],\end{aligned}$$

where distribution Q_ψ is the approximate posterior distribution with parameters ψ, computed similarly as the prior distribution but further conditioned on the *encoder* RNN hidden state of the next utterance.

The original *VHRED* model (Serban et al., 2017b) used Gaussian latent variables. We refer to this model as *G-VHRED*. The first model we propose uses piecewise constant latent variables instead of Gaussian latent variables. We refer to this model as *P-VHRED*. The second model we propose takes advantage of the representation power of both Gaussian and piecewise constant latent variables. This model samples both a Gaussian latent variable z_n^{gaussian} and a piecewise latent variable $z_n^{\text{piecewise}}$ independently conditioned on the *context* RNN hidden state:

$$\begin{aligned}P_\theta(z_n^{\text{gaussian}} \mid \mathbf{w}_{<n}) &= f_\theta^{\text{prior, gaussian}}(z_n^{\text{gaussian}}; h_{n-1}^{con}),\\ P_\theta(z_n^{\text{piecewise}} \mid \mathbf{w}_{<n}) &= f_\theta^{\text{prior, piecewise}}(z_n^{\text{piecewise}}; h_{n-1}^{con}),\end{aligned}$$

where $f^{\text{prior, gaussian}}$ and $f^{\text{prior, piecewise}}$ are PDFs parametrized by independent subsets of parameters θ. We refer to this model as *H-VHRED*.

6 Experiments

We evaluate the proposed models on two types of natural language processing tasks: document modeling and dialogue natural language generation. All models are trained with back-propagation using the variational lower-bound on the log-likelihood or the exact log-likelihood. We use the first-order gradient descent optimizer Adam (Kingma and Ba, 2015) with gradient clipping (Pascanu et al., 2012)[1]

Model	**20-NG**	**RCV1**	**CADE**
LDA	1058	−−	−−
docNADE	896	−−	−−
NVDM	836	−−	−−
G-NVDM	651	905	339
H-NVDM-3	607	865	**258**
H-NVDM-5	**566**	**833**	294

Table 1: Test perplexities on three document modeling tasks: 20-NewGroup (20-NG), Reuters corpus (RCV1) and CADE12 (CADE). Perplexities were calculated using 10 samples to estimate the variational lower-bound. The *H-NVDM* models perform best across all three datasets.

6.1 Document Modeling

Tasks We use three different datasets for document modeling experiments. First, we use the 20 News-Groups (20-NG) dataset (Hinton and Salakhutdinov, 2009). Second, we use the Reuters corpus (RCV1-V2), using a version that contained a selected 5,000 term vocabulary. As in previous work (Hinton and Salakhutdinov, 2009; Larochelle and Lauly, 2012), we transform the original word frequencies using the equation $\log(1 + \text{TF})$, where TF is the original word frequency. Third, to test our document models on text from a non-English language, we use the Brazilian Portuguese CADE12 dataset (Cardoso-Cachopo, 2007). For all datasets, we track the validation bound on a subset of 100 vectors randomly drawn from each training corpus.

Training All models were trained using mini-batches with 100 examples each. A learning rate of 0.002 was used. Model selection and early stopping were conducted using the validation lower-bound, estimated using five stochastic samples per validation example. Inference networks used 100 units in each hidden layer for 20-NG and CADE, and 100 for RCV1. We experimented with both 50 and 100 latent random variables for each class of models, and found that 50 latent variables performed best on the validation set. For *H-NVDM* we vary the number of components used in the PDF, investigating the effect that 3 and 5 pieces had on the final quality of the model. The number

[1] Code and scripts are available at `https://github.com/ago109/piecewise-nvdm-emnlp-2017` and `https://github.com/julianser/hred-latent-piecewise`.

G-NVDM	H-NVDM-3	H-NVDM-5
environment	project	science
project	gov	built
flight	major	high
lab	based	technology
mission	earth	world
launch	include	form
field	science	scale
working	nasa	sun
build	systems	special
gov	technical	area

Table 2: Word query similarity test on 20 News-Groups: for the query 'space", we retrieve the top 10 nearest words in word embedding space based on Euclidean distance. *H-NVDM-5* associates multiple meanings to the query, while *G-NVDM* only associates the most frequent meaning.

of hidden units was chosen via preliminary experimentation with smaller models. On 20-NG, we use the same set-up as (Hinton and Salakhutdinov, 2009) and therefore report the perplexities of a topic model (*LDA*, (Hinton and Salakhutdinov, 2009)), the document neural auto-regressive estimator (*docNADE*, (Larochelle and Lauly, 2012)), and a neural variational document model with a fixed standard Gaussian prior (*NVDM*, lowest reported perplexity, (Miao et al., 2016)).

Results In Table 1, we report the test document perplexity: $\exp(-\frac{1}{D}\sum_n \frac{1}{L_n} \log P_\theta(x_n)$. We use the variational lower-bound as an approximation based on 10 samples, as was done in (Mnih and Gregor, 2014). First, we note that the best baseline model (i.e. the *NVDM*) is more competitive when both the prior and posterior models are learnt together (i.e. the *G-NVDM*), as opposed to the fixed prior of (Miao et al., 2016). Next, we observe that integrating our proposed piecewise variables yields even better results in our document modeling experiments, substantially improving over the baselines. More importantly, in the 20-NG and Reuters datasets, increasing the number of pieces from 3 to 5 further reduces perplexity. Thus, we have achieved a new state-of-the-art perplexity on 20 News-Groups task and — to the best of our knowledge – better perplexities on the CADE12 and RCV1 tasks compared to using a state-of-the-art model like the *G-NVDM*. We also evaluated the converged models using an non-parametric inference procedure, where a separate approximate posterior is learned for each test example in order to tighten the variational lower-bound. *H-NVDM* also performed best in this evaluation across all three datasets, which confirms that the performance improvement is due to the piecewise components. See appendix for details.

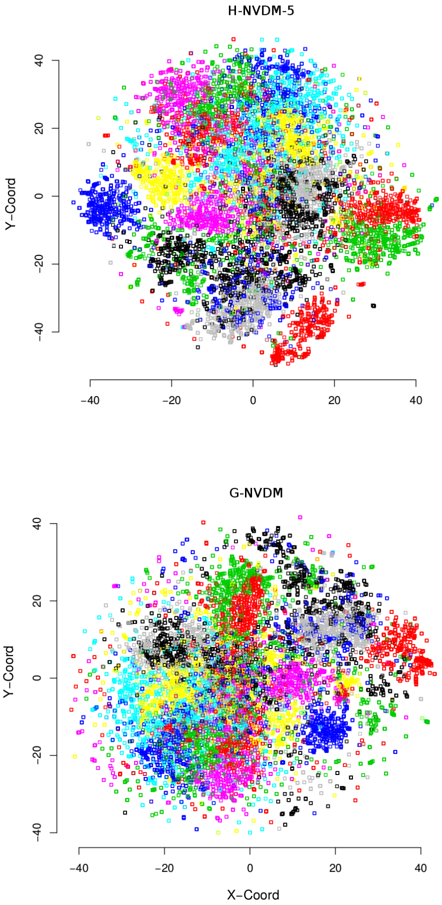

Figure 2: Latent variable approximate posterior means t-SNE visualization on 20-NG for *G-NVDM* and *H-NVDM-5*. Colors correspond to the topic labels assigned to each document.

In Table 2, we examine the top ten highest ranked words given the query term "space", using the decoder parameter matrix. The piecewise variables appear to have a significant effect on what is uncovered by the model.In the case of "space", the hybrid with 5 pieces seems to value two senses of the word–one related to "outer space" (e.g., "sun", "world", etc.) and another related to the dimensions of depth, height, and width within which things may exist and move (e.g., "area", "form", "scale", etc.). On the other hand, *G-NVDM* appears to only capture the "outer space" sense of

Model	Activity	Entity
HRED	4.77	2.43
G-VHRED	**9.24**	2.49
P-VHRED	5	2.49
H-VHRED	8.41	**3.72**

Table 3: Ubuntu evaluation using F1 metrics w.r.t. activities and entities. *G-VHRED*, *P-VHRED* and *H-VHRED* all outperform the baseline *HRED*. *G-VHRED* performs best w.r.t. activities and *H-VHRED* performs best w.r.t. entities.

the word. More examples are in the appendix.

Finally, we visualized the means of the approximate posterior latent variables on 20-NG through a t-SNE projection. As shown in Figure 2, both *G-NVDM* and *H-NVDM-5* learn representations which disentangle the topic clusters on 20-NG. However, *G-NVDM* appears to have more dispersed clusters and more outliers (i.e. data points in the periphery) compared to *H-NVDM-5*. Although it is difficult to draw conclusions based on these plots, these findings could potentially be explained by the Gaussian latent variables fitting the latent factors poorly.

6.2 Dialogue Modeling

Task We evaluate *VHRED* on a natural language generation task, where the goal is to generate responses in a dialogue. This is a difficult problem, which has been extensively studied in the recent literature (Ritter et al., 2011; Lowe et al., 2015; Sordoni et al., 2015; Li et al., 2016; Serban et al., 2016a,b). Dialogue response generation has recently gained a significant amount of attention from industry, with high-profile projects such as Google SmartReply (Kannan et al., 2016) and Microsoft Xiaoice (Markoff and Mozur, 2015). Even more recently, Amazon has announced the Alexa Prize Challenge for the research community with the goal of developing a natural and engaging chatbot system (Farber, 2016).

We evaluate on the technical support response generation task for the Ubuntu operating system. We use the well-known Ubuntu Dialogue Corpus (Lowe et al., 2015, 2017), which consists of about 1/2 million natural language dialogues extracted from the #Ubuntu Internet Relayed Chat (IRC) channel. The technical problems discussed span a wide range of software-related and hardware-related issues. Given a dialogue history — such as a conversation between a user and a technical support assistant — the model must generate the next appropriate response in the dialogue. For example, when it is the turn of the technical support assistant, the model must generate an appropriate response helping the user resolve their problem.

We evaluate the models using the activity- and entity-based metrics designed specifically for the Ubuntu domain (Serban et al., 2017a). These metrics compare the *activities* and *entities* in the model generated responses with those of the reference responses; activities are verbs referring to high-level actions (e.g. *download*, *install*, *unzip*) and entities are nouns referring to technical objects (e.g. *Firefox*, *GNOME*). The more activities and entities a model response overlaps with the reference response (e.g. expert response) the more likely the response will lead to a solution.

Training The models were trained to maximize the log-likelihood of training examples using a learning rate of 0.0002 and mini-batches of size 80. We use a variant of truncated backpropagation. We terminate the training procedure for each model using early stopping, estimated using one stochastic sample per validation example. We evaluate the models by generating dialogue responses: conditioned on a dialogue context, we fix the model latent variables to their median values and then generate the response using a beam search with size 5. We select model hyperparameters based on the validation set using the F1 activity metric, as described earlier.

It is often difficult to train generative models for language with stochastic latent variables (Bowman et al., 2015; Serban et al., 2017b). For the latent variable models, we therefore experiment with reweighing the KL divergence terms in the variational lower-bound with values 0.25, 0.50, 0.75 and 1.0. In addition to this, we linearly increase the KL divergence weights starting from zero to their final value over the first 75000 training batches. Finally, we weaken the *decoder* RNN by randomly replacing words inputted to the decoder RNN with the unknown token with 25% probability. These steps are important for effectively training the models, and the latter two have been used in previous work by Bowman et al. (2015) and Serban et al. (2017b).

HRED (Baseline): We compare to the *HRED* model (Serban et al., 2016a): a sequence-to-sequence model, shown to outperform other established models on this task, such as the LSTM RNN language model (Serban et al., 2017a). The *HRED* model's *encoder* RNN uses a bidirectional GRU RNN encoder, where the forward and backward RNNs each have 1000 hidden units. The context RNN is a GRU encoder with 1000 hidden units, and the decoder RNN is an LSTM decoder with 2000 hidden units.[2] The encoder and context RNNs both use layer normalization (Ba et al., 2016).[3] We also experiment with an additional rectified linear layer applied on the inputs to the decoder RNN. As with other hyper-parameters, we choose whether to include this additional layer based on the validation set performance. *HRED*, as well as all other models, use a word embedding dimensionality of size 400.

G-HRED: We compare to *G-VHRED*, which is *VHRED* with Gaussian latent variables (Serban et al., 2017b). *G-VHRED* uses the same hyper-parameters for the encoder, context and decoder RNNs as the HRED model. The model has 100 Gaussian latent variables per utterance.

P-HRED: The first model we propose is *P-VHRED*, which is *VHRED* model with piecewise constant latent variables. We use $n = 3$ number of pieces for each latent variable. *P-VHRED* also uses the same hyper parameters for the encoder, context and decoder RNNs as the *HRED* model. Similar to *G-VHRED*, *P-VHRED* has 100 piecewise constant latent variables per utterance.

H-HRED: The second model we propose is *H-VHRED*, which has 100 piecewise constant (with $n = 3$ pieces per variable) and 100 Gaussian latent variables per utterance. *H-VHRED* also uses the same hyper-parameters for the encoder, context and decoder RNNs as *HRED*.

Results: The results are given in Table 3. All latent variable models outperform *HRED* w.r.t. both activities and entities. This strongly suggests that the high-level concepts represented by the latent variables help generate meaningful, goal-directed responses. Furthermore, each type of latent variable appears to help with a different aspects of the generation task. *G-VHRED* performs best w.r.t. activities (e.g. *download*, *install* and so on), which occur frequently in the dataset. This suggests that the Gaussian latent variables learn useful latent representations for frequent actions. On the other hand, *H-VHRED* performs best w.r.t. entities (e.g. *Firefox*, *GNOME*), which are often much rarer and mutually exclusive in the dataset. This suggests that the combination of Gaussian and piecewise latent variables help learn useful representations for entities, which could not be learned by Gaussian latent variables alone. We further conducted a qualitative analysis of the model responses, which supports these conclusions. See Appendix G.[4]

7 Conclusions

In this paper, we have sought to learn rich and flexible multi-modal representations of latent variables for complex natural language processing tasks. We have proposed the piecewise constant distribution for the variational autoencoder framework. We have derived closed-form expressions for the necessary quantities required for in the autoencoder framework, and proposed an efficient, differentiable implementation of it. We have incorporated the proposed piecewise constant distribution into two model classes — *NVDM* and *VHRED* — and evaluated the proposed models on document modeling and dialogue modeling tasks. We have achieved state-of-the-art results on three document modeling tasks, and have demonstrated substantial improvements on a dialogue modeling task. Overall, the results highlight the benefits of incorporating the flexible, multi-modal piecewise constant distribution into variational autoencoders. Future work should explore other natural language processing tasks, where the data is likely to arise from complex, multi-modal latent factors.

Acknowledgments

The authors acknowledge NSERC, Canada Research Chairs, CIFAR, IBM Research, Nuance Foundation and Microsoft Maluuba for funding. Alexander G. Ororbia II was funded by a NACME-Sloan scholarship. The authors thank Hugo Larochelle for sharing the NewsGroup 20 dataset. The authors thank Laurent Charlin, Sungjin Ahn, and Ryan Lowe for constructive feedback. This research was enabled in part by support provided by Calcul Qubec (www.calculquebec.ca) and Compute Canada (www.computecanada.ca).

[2] Since training lasted between 1-3 weeks for each model, we had to fix the number of hidden units during preliminary experiments on the training and validation datasets.

[3] We did not apply layer normalization to the decoder RNN, because several of our colleagues have found that this may hurt the performance of generative language models.

[4] Results on a Twitter dataset are given in the appendix.

References

J. L. Ba, J. R. Kiros, and G. E. Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

S. Bangalore, G. Di Fabbrizio, and A. Stent. 2008. Learning the structure of task-driven human–human dialogs. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1249–1259.

D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent dirichlet allocation. *JAIR*, 3:993–1022.

J. Bornschein and Y. Bengio. 2015. Reweighted wake-sleep. In *ICLR*.

S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. 2015. Generating sentences from a continuous space. In *Conference on Computational Natural Language Learning*.

Y. Burda, R. Grosse, and R. Salakhutdinov. 2016. Importance weighted autoencoders. *ICLR*.

A. Cardoso-Cachopo. 2007. Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa.

X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. 2017. Variational lossy autoencoder. In *ICLR*.

N. Crook, R. Granell, and S. Pulman. 2009. Unsupervised classification of dialogue acts using a dirichlet process mixture model. In *Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 341–348.

P. Dayan and G. E. Hinton. 1996. Varieties of helmholtz machine. *Neural Networks*, 9(8):1385–1403.

L. Devroye. 1986. Sample-based non-uniform random variate generation. In *Proceedings of the 18th conference on Winter simulation*, pages 260–265. ACM.

M. Farber. 2016. Amazon's 'Alexa Prize' Will Give College Students Up To $2.5M To Create A Socialbot. *Fortune*.

M. Fraccaro, S. K. Sønderby, U. Paquet, and O. Winther. 2016. Sequential neural models with stochastic layers. In *NIPS*, pages 2199–2207.

K. Gregor, I. Danihelka, A. Graves, and D. Wierstra. 2015. DRAW: A recurrent neural network for image generation. In *ICLR*.

G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. 1995. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161.

G. E. Hinton and R. Salakhutdinov. 2009. Replicated softmax: an undirected topic model. In *NIPS*, pages 1607–1614.

G. E. Hinton and R. S. Zemel. 1994. Autoencoders, minimum description length and helmholtz free energy. In *NIPS*, pages 3–10. NIPS.

T. Hofmann. 1999. Probabilistic latent semantic indexing. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM.

E. Jang, S. Gu, and B. Poole. 2017. Categorical reparameterization with gumbel-softmax. In *ICLR*.

M. Johnson, D. K. Duvenaud, A. Wiltschko, R. P. Adams, and S. R. Datta. 2016. Composing graphical models with neural networks for structured representations and fast inference. In *NIPS*, pages 2946–2954.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. 1999. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

A. Kannan, K. Kurach, et al. 2016. Smart Reply: Automated Response Suggestion for Email. In *KDD*.

D. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

D. P. Kingma, T. Salimans, and M. Welling. 2016. Improving variational inference with inverse autoregressive flow. *NIPS*, pages 4736–4744.

D. P. Kingma and M. Welling. 2014. Auto-encoding variational Bayes. *ICLR*.

H. Larochelle and S. Lauly. 2012. A neural autoregressive topic model. In *NIPS*, pages 2708–2716.

A. B. Lindbo Larsen, S. K. Sønderby, and O. Winther. 2016. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, pages 1558–1566.

S. Lauly, Y. Zheng, A. Allauzen, and H. Larochelle. 2016. Document neural autoregressive distribution estimation. *arXiv preprint arXiv:1603.05962*.

J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *The North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 110–119.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. In *Special Interest Group on Discourse and Dialogue (SIGDIAL)*.

Ryan T. Lowe, Nissan Pow, Iulian V. Serban, Laurent Charlin, Chia-Wei Liu, and Joelle Pineau. 2017. Training End-to-End Dialogue Systems with the Ubuntu Dialogue Corpus. *Dialogue & Discourse*, 8(1).

Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. 2016. Auxiliary deep generative models. In *ICML*, pages 1445–1453.

C. J. Maddison, A. Mnih, and Y. W. Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*.

J. Markoff and P. Mozur. 2015. For Sympathetic Ear, More Chinese Turn to Smartphone Program. *New York Times*.

Y. Miao, L. Yu, and P. Blunsom. 2016. Neural variational inference for text processing. In *ICML*, pages 1727–1736.

A. Mnih and K. Gregor. 2014. Neural variational inference and learning in belief networks. In *ICML*, pages 1791–1799.

R. M. Neal. 1992. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113.

A. G. Ororbia II, C. L. Giles, and D. Reitter. 2015. Online semi-supervised learning with deep hybrid boltzmann machines and denoising autoencoders. *arXiv preprint arXiv:1511.06964*.

R. Pascanu, T. Mikolov, and Y. Bengio. 2012. On the difficulty of training recurrent neural networks. *ICML*, 28:1310–1318.

Rajesh Ranganath, Dustin Tran, and David Blei. 2016. Hierarchical variational models. In *ICML*, pages 324–333.

D. J. Rezende and S. Mohamed. 2015. Variational inference with normalizing flows. In *ICML*, pages 1530–1538.

D. J. Rezende, S. Mohamed, and D. Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286.

A. Ritter, C. Cherry, and W. B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 583–593.

J. T. Rolfe. 2017. Discrete variational autoencoders. In *ICLR*.

Francisco R Ruiz, Michalis Titsias RC AUEB, and David Blei. 2016. The generalized reparameterization gradient. In *NIPS*, pages 460–468.

R. Salakhutdinov and G. E. Hinton. 2009. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978.

R. Salakhutdinov and H. Larochelle. 2010. Efficient learning of deep boltzmann machines. In *AISTATs*, pages 693–700.

T. Salimans, D. P Kingma, and M. Welling. 2015. Markov chain monte carlo and variational inference: Bridging the gap. In *ICML*, pages 1218–1226.

R. Sennrich, B. Haddow, and A. Birch. 2016. Neural machine translation of rare words with subword units. In *Association for Computational Linguistics (ACL)*.

I. V. Serban, T. Klinger, G. Tesauro, K. Talamadupula, B. Zhou, Y. Bengio, and A. Courville. 2017a. Multiresolution recurrent neural networks: An application to dialogue response generation. In *Thirty-First AAAI Conference (AAAI)*.

I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. 2016a. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference (AAAI)*.

I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, and Y. Bengio. 2017b. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference (AAAI)*.

Iulian Vlad Serban, Ryan Lowe, Laurent Charlin, and Joelle Pineau. 2016b. Generative deep neural networks for dialogue: A short review. In *NIPS, Let's Discuss: Learning Methods for Dialogue Workshop*.

A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J. Nie, J. Gao, and B. Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2015)*, pages 196–205.

N. Srivastava, R. R Salakhutdinov, and G. E. Hinton. 2013. Modeling documents with deep boltzmann machines. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 616–624.

B. Uria, I. Murray, and H. Larochelle. 2014. A deep and tractable density estimator. In *ICML*, pages 467–475.

K. Zhai and J. D. Williams. 2014. Discovering latent structure in task-oriented dialogues. In *Association for Computational Linguistics (ACL)*, pages 36–46.