Disentangling latents in a Variational RNN

Abishek Prasanna *
Department of Computer Science
Rutgers University
abishekprasanna@gmail.com

Manish Vidyasagar
Department of Computer Science
Rutgers University
mv549@scarletmail.rutgers.edu

Abstract

In this paper, we explore the introduction of latent variables in stochastic recurrent neural networks with the aim of disentangling factors of variation in image generation. Our deep generative model extends the variational RNN J. Chung [2015] model with a principled framework that allows us to approximately disentangle style and content from sequence of images. In our experiments, we analyze the benefits of using stochastic RNN models for disentanglement and also draw insights on some of the core challenges that need to be addressed in order to achieve further success in disentangled representation learning. Code of our algorithm maybe accessed at https://github.com/Abishekpras/vrnn

1 Introduction

Deep generative models have shown much promise especially in the area of representation learning because of their ability to capture high level concepts via latent variable modeling. However, learning interpretable factorized representations from data still remains an outstanding research problem in machine learning. The goal of disentangled representation learning is to capture the independent data generative factors of the world, and it is thought to be an important precursor for Artificial General Intelligence(AGI) that can learn and think like humans. Models applied to Computer Vision tasks, do not fare very well in teasing out the data generative factors in images. Learning a disentangled representation for images should ideally allow the model to capture a content code that represents the geometrical information of the data, and a style code that captures its textural properties. Models that can capture such data generative properties may find a lot of application in data compression, style transfer and many other downstream tasks.

Learning disentangled style and content codes is a challenging task. Several algorithms have been developed for solving this class of problems. Mathieu [2016] investigated using variational autoencoders to partition the encoding space into style and content components, and performed adversarial training to encourage the datapoints from the same class to have similar content representations, but diverse style features. H.Kazemi [2019] extend GAN framework to this problem, and are able to decompose the latent into style and content codes for each image. Y. [2018] demonstrated the model of a Disentangled sequential autoencoder that learns to disentangle content and style from sequence data, and their work is further evidence for the hypothesis that stochastic RNNs as latent state models are powerful enough at compressing and generating long sequences.

J. Chung [2015] proposed the Variational RNN model that used high level latent random variables that could model the kind of variability that is observed in highly structured sequential data. Although VRNN's are powerful generative models, they are not aimed at disentangled representation learning and do not explicitly disentangle the representation of time-invariant and time-dependent information. Research in generative modeling can therefore be categorized into two sub-types. 1)

*Website: https://abishekpras.github.io

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Generative models that can learn high-dimensional data and model long-range dependencies and can produce state-of-the-art results in modelling the data and generate accurate reconstructions, and 2) Generative models that aim to learn interpretable and factorized representations from data, and thus capture to some extent the data generative factors of the real world. In this paper, our investigation is primarily aimed at understanding this divide in generative modeling and list out the challenges that need to be addressed in order to bridge the differences in performance (data generation/sharpness/accuracy) and (interpretable) representation learning. Inspired by the work of J. Chung [2015], we aim to use powerful generative models of stochastic RNNs and learn content and style disentanglement from image sequence data. In more detail, our contributions are as follows:

1 Disentangled_VRNN: We show how Variational RNN maybe extended by adding more latent random variables to capture disentangled content and style codes from image sequence data.

2 Research Proposals Experiments with our model provides fresh insights on potential research directions that need to be pursued for success in content-style disentanglement from image sequence data.

The paper is structured as follows. Section 2 introduces the generative model and the problem setting. Section 3 discusses related work and differences from our model. Section 4 presents three experiments on MNIST image sequence data. Finally, Section 5 concludes the paper and discusses future research directions.

2 Disentangled VRNN model

In this section, we introduce our Fully Disentangled VRNN model. Our model is adapted from ideas in J. Chung [2015] and also Y. [2018] and is a deep generative model designed as a stochastic RNN with latent variables to model the time-dependent and time-independent data generative factors of variation.

Generation: The VAEs at every time-step of the VRNN are conditioned on the state variable h_{t-1} of an RNN. Unlike a standard VAE, the prior on the latent random variable is no longer a standard Gaussian distribution, but follows the distribution N($\mu_{(0,t)}, \sigma_{(0,t)}$) where $(\mu_{(0,t)}, \sigma_{(0,t)}) \sim \psi_T^{prior}(h_{t-1})$
The content latent variable f is sampled from a standard normal distribution, and is passed through every time-step of the RNN. Within every time-step we run a VAE model setup wherein we learn mean and standard deviation and reparameterize to sample a latent style code. The complete generative model is as follows:

$$p_\theta(z_{1:T}, f, x_{1:T}) = p_{N(0,1)}(f) \prod_{t=1}^{T} p_\theta(z_t|z_{<t}, f, x_{<t}) p_\theta(x_t|z_{\leq t, f, x_{<t}})$$

Inference: In a similar fashion we infer the style code z as conditioned on the previous hidden state h_{t-1}. But before this, we infer the content code f as the last output of a RNN with all the inputs in the sequence. This will allow the model to capture time-independent factors in f, and ideally force z to pick up dynamic factors of variability. At each time step, we pass in the learned content code f in order to infer the next hidden state of the recurrent neural net. This mechanism can be thought of like a skip-connection, wherein the other inputs to h_t will have to learn everything that is the residual of the information being passed on by the content code f. The complete inference model is as follows:

$$q_\phi(z_{1:T}, f|x_{1:T}) = q_\phi(f|x_{1:T}) \prod_{t=1}^{T} q_\phi(z_t|z_{<t}, x_{\leq t})$$

Learning: The objective function includes three components. A KL divergence term for the content code f which is computed as follows:

$$KL_f = -KL(q(f|x_{1:T})||p(f))$$

This is similar to a VAE except for the fact that the input here is a sequence and therefore uses a RNN to learn the code f instad of a feedforward neural network.

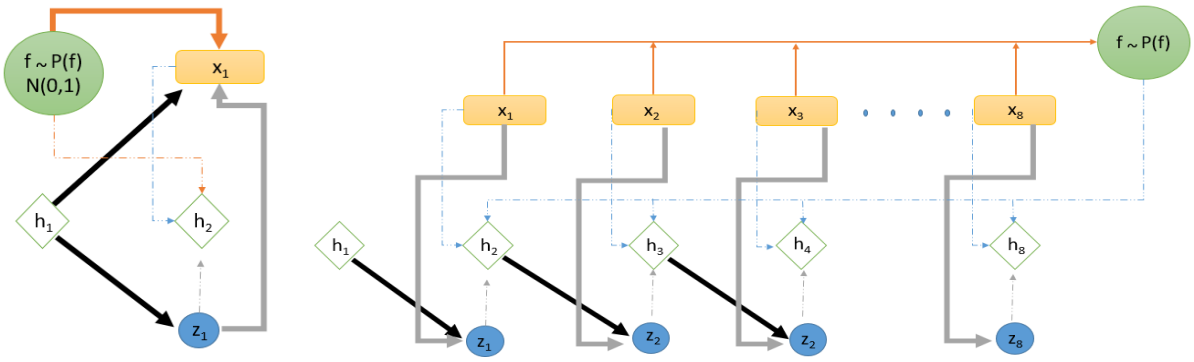

Figure 1: (Left) Generation model for a single time step. (Right) Inference of latent codes f and $z_{1:T}$ from data points $X_{1:T}$. Solid lines represent the main directions of data flow for generation and inference, respectively. Dashed lines represent the flow of information from one time-step to the next in the process of generating or inferring for data of T time-steps.

The time-varying latent code z is computed in a similar fashion as to what we have seen in Variational RNN model as a time-step wise KL term, as:

$$KL_z = \sum_{t=1}^{T} -KL(q(z_t|x_{\leq t}, z_{<t})||p(z_t|x_{<t}, z_{<t}))$$

The reconstruction loss term remains the same as in the traditional VAE setup and is computed as a cross-entropy loss.

$$recon\_loss = \sum_{t=1}^{T} log\, p(x_t|z_{\leq t}, x_{<t})$$

Therefore, adding up all of these error terms we get the complete loss function as:

$$E_{q(z_{1:T},f|x_{1:T})}[\beta_1 * KL_f + \beta_2 * KL_z + recon\_loss]$$

3 Related Work

3.1 Beta-VAE

The variational autoencoder (VAE) D.Kingma [2014] is a latent variable model that pairs a top-down generator with a bottom-up inference network. Instead of directly performing maximum likelihood estimation on the intractable marginal log-likelihood, training is done by optimizing the tractable evidence lower bound (ELBO). We would like to optimize this lower bound averaged over the empirical distribution (with β = 1):

$$L_\beta = \frac{1}{N} \sum_{n=1}^{N} (E_q(log\ \ p(x_n|z)) - \beta KL(q(z|x_n)||p(z)))$$

The β-VAE Higgins [2017] is a variant of the variational autoencoder that attempts to learn a disentangled representation by optimizing a heavily penalized objective with $\beta > 1$. Such simple penalization has been shown to be capable of obtaining models with a high degree of disentanglement in image datasets. However, it is not made explicit why penalizing $KL(q(z|x)||p(z))$ with a factorial prior can lead to learning latent variables that exhibit disentangled transformations for all data samples.

3.1.1 Variational RNN

VRNN was the first proposed model incorporating a VAE at every time step of an RNN, that aimed to learn good prior distributions for latent random variables by conditioning on all previous inputs.

Figure 2: Training data for Disentangled_vrnn are fetched as mini-batches of 8 images each from a particular class of MNIST digits. They are sampled and independently and each image therefore has the same content but differing magnitudes of style factors like "angle/orientation", "thickness" etc.

Furthermore, they introduced stochasticity in the transition function of an RNN, in addition to the RNN output stochasticity. VRNN demonstrated good performance in highly structured sequence data by explicitly modeling the dependency between latent random variables across time-steps and thereby taking int account the temporal structure of the sequential data. Their generative model is given by:

$$p(x_{\leq T}, z_{\leq T}) = \prod_{t=1}^{T} p(x_t|z_{\leq t}, x_{<t})p(z_t|x_{<t}, z_{<t})$$

3.1.2 Disentangled Sequential Autoencoder

Y. [2018] demonstrated a generative model for learning disentangled representations of a high-dimensional time series. They explore stochastic RNN models to capture disentanglement of content amd style n video frame data which is much more structured and ordered than independently sampled MNIST data sequence. They however use models which are conditioned on the previous vaues alone, and condition only on z_{t-1} and x_{t-1}. Their full generative model is given as follows:

$$q_\phi(z_{1:T}, f|x_{1:T}) = q_\phi(f|x_{1:T})q_\phi(z_{1:T}|f, x_{1:T})$$

4 Experiments

We implemented a basic VRNN model as proposed by J. Chung [2015] with PyTorch to train on randomly sampled sequences of MNIST images. We further extended it to our disentangled VRNN model, and adopted most of the ideas from J. Chung [2015] like having special feature extractors for data and latent codes. We used a GRU model for both the RNNs encoding content and style. We ran all experiments on MNIST data and considered data to come as sequence from the same class, when running our disentangled VRNN experiments in the format shown in Figure 1. We used gradient clipping and trained the model for 100 epochs with the standard training set of MNIST data.

For evaluation, we relied on qualitative analysis and latent space interpolation experiments as is common is the disentangled representation learning literature. We did not do any quantitative analysis on our model because MNIST dataset does not come with the metrics representing the generative factor values that are traditionally used by models to train a classifier and quantitatively measure and analyze the disentanglement metric as was done in Higgins [2017]. We were able to sample and conditionally generate images while fixing either of content or style codes, and those outputs are shown in Figure 4(best results) and 5(some failure examples).

We performed three key experiments, to assess our model and analyze the challenges and other issues that hinder disentangled representation learning in our model.

4.1 Beta-VAE on MNIST

When we began our investigation of content-style disentangling models, we had a focus on keeping a model as extensible to the continual learning regime, so that it may resemble how humans learn independent concepts incrementally. To the best of our knowledge, there has not been any previous work that accommodates for disentangled representation learning in a continual learning regime.

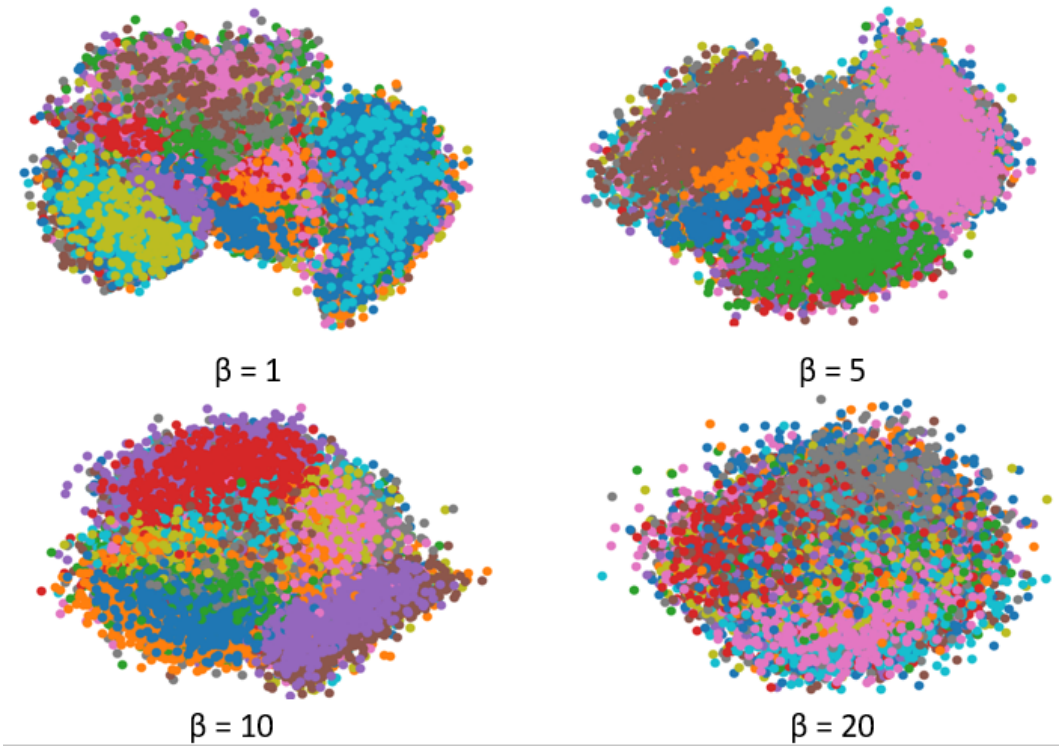

Figure 3: Beta-VAE(beta=1,5,10,20) disentanglement of latent code space of MNIST digits when learning a 2-dimensional latent representation.

Higgins [2017] Beta VAE and the related family of models like Beta-TCVAE Chen [2018], Factor VAE H.Kim [2018] are examples of models that learn disentangled representations with a VAE from a single image, as opposed to sequence of images like in Y. [2018] Disentangled Sequential Autoencoder or in our disentangled VRNN. Learning such factors of variation from a single image makes it much more feasible to apply continual learning on these models, and thus we experimented on extending a Beta-VAE model to address disentangled content and style representation learning.

We trained a beta-VAE model with continual capacity increment on MNIST dataset, and analyzed its latent space to verify if the space is neatly separated with respect to either content or style. We performed a qualitative analysis by running k-nearest neighbours on the learned 2-dimensional latent representation, and visualizing the samples generated by the nearest latent vectors. In our analysis we could not find much similarity in content or style being captured in the latent space, and the clustering space learned was not neat enough and hugely cluttered with other-class codes interspersed as shown in Figure 3.

We were not able to extend the beta-VAE model to accommodate time-dependent and time-independent codes from a single image, and it would be interesting to see models that can achieve such results in the future. This experiment led us to pursuing the proposed disentangled VRNN model that has sequence of images as input data and uses stochastic RNNs that can observe and identify time-varying features from the image sequence.

4.2 Hyperparameter sensitivity for disentanglement

Inspired by the ideas adopted in Beta-VAE for learning independent factors by scaling up the KL divergence terms in the variational lower bound, we adopt this similar mechanism in the disentangled_VRNN model. A VRNN is essentially a VAE at every time step, and so this kind of weighting can be adopted very naturally into our model. In our experiments with different hyperparameter settings coupled with a qualitative analysis of disentanglement, we were able to observe that our

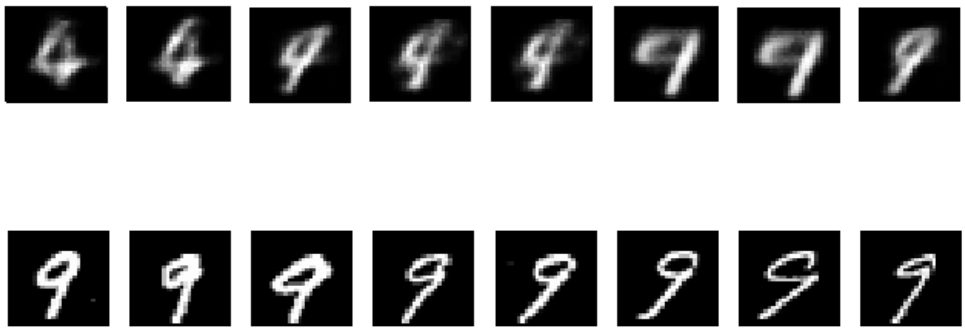

Figure 4: Controlled generation of image samples from random noise. (Top) We sample a single style variable z and use it across all 8 timesteps, while randomly sampling content variable f each time. Note the smooth transition of content across timesteps which maybe due to the strong prior learned by VRNN as opposed to the network actually learning content code in f. There is further evidence for this, in the fact that we were able to obtain only a poor reconstruction when sampling with fixed style code z, as opposed to fixed content. (Bottom) We take a single sample of content code f while randomly sampling style code z in each time-step. Here again, the style transition seems very smooth which suggests that the strong prior learned by VRNN maybe passing on more information thereby not forcing the network to encode properties accurately.

model is highly sensitive to the hyperparameter settings. We independently weighted the content and style KL terms and had the most optimal results when $\beta_f = 0.5$ and $\beta_z = 25$. The reconstructions obtained were also found to be less clear and sharp (corresponding to high likelihood loss) for different hyperparameter values.

4.3 Content-Style independence assumptions

In addition to the proposed model [Full disentangled VRNN] in Section 2, we add another model which we call [Factorized disentangled VRNN] which assumes independence on content and style codes. One major factor in deciding these independence assumptions was the idea that the training data showed examples of images/digits that were structurally slightly different when adopting different styles. Figure 2 shows generated image of the digit 9 with different style, and it can be seen that the stretch of the digit varied according to the angle in which it was oriented in most of the dataset. Another way of thinking about this would be to imagine how the freedom of motion of our body influences the dynamics of motion and thereby imposing conditions on the style. Some of the example samples derived from Factorized VRNN is shown in Figure 5., and a comparison with Fully Disentangled VRNN, on the KL Divergence values upon convergence is provided in Table 6 in the Appendix.

5 Conclusion

We presented a deep generative model for learning disentangled representations of high-dimensional image sequence data. Our model consists of a global latent variable for content features, and a stochastic RNN with time-local latent variables for dynamical features, that leverages a stochastic transition function as in VRNN. The model is trained using standard amortized variational inference. We carried out experiments on MNIST data. Our approach allows us to perform full and conditional generation.

Although disentangled VRNN is able to demonstrate controlled generation of samples with variability in content and style, one at a time, further experimentation and analysis gives us a lot of useful hints that help understand the model. VRNN is a powerful generative model that conditions

the latents and generated data on all the previous latents and data, which may not be aiding our cause. Y. [2018] were able to demonstrate content-style disentanglement without conditioning on all previous z_t and x_t, and instead only on the previous values. This could also be because the data they considered were video frames and s had a particular sequential form to it. However, our qualitative analysis suggests that the content latent variable may not be learning much useful information and this information is being passed on by the encoded hidden state or entangled with the style latent variables. We arrive at this conclusion from the analysis of the generated samples and the difference in likelihood loss that is evident between models optimized for one factor's controlled generation over the other by enforcing stronger hyperparameters, as shown in Fig 3.

However, this does not mean that VRNN model cannot be extended efficiently to disentangled representation learning. We note some of the opportunities of improvement that come out of our model architecture. As opposed to Y. [2018], our model conditions the hidden state of the recurrent network instead of the style code directly. This allows us to have two recurrent networks one for learning the style code and the other for the content. In our literature survey we found a few models that use Analogy based visual reasoning to produce disentanglement like D.Kingma [2015]. Similar approaches could be adapted the disentangled VRNN model to make it more robust. One approach would be using Attention mechanism to compare and pick out the most relevant content encoding hidden-state instead of always choosing the summary of all hidden states when conditioning our VRNN at every time-step. Our model is then very much similar in setup to Sequence-to-Sequence models made popular by I.Sutskever [2014] and which have shown to have considerably improved performance when using attention schemes. We believe such improvements maybe introduced in our disentangled VRNN framework as well.

One of the major challenges we faced in working on this problem, is the problem of learning good representations for the content variable. By nature of the model it is clear that there aren't any informative priors to base our assumptions on, and having a solution for this would greatly mitigate most of our concerns. Research in introducing Bayesian non-parametric priors to neural networks is still at a nascent stage but holds significant promise in this regard. Our future direction of research is to explore adding informative priors of the like being used in W.Joo [2019] Dirichlet VAE and E.Nalisnick [2017] Stick-Breaking VAE into our model with an idea to learn good priors to tackle the problem of learning good adaptive representations that capture real-world concepts.

Acknowledgments

We would like to thank Prof.Sungjin Ahn for his guidance, support and for providing the opportunity to explore this topic for our course project as part of CS671 course at Rutgers University.

Contribution

Ideation and Literature Review :	Abishek Prasanna, Manish Vidyasagar
Beta-VAE experiments :	Manish Vidyasagar
VRNN and Disentangled VRNN impl. & expts :	Abishek Prasanna
Evaluation :	Abishek Prasanna
Presentation :	Abishek Prasanna, Manish Vidyasagar
Documentation/Report Writing :	Abishek Prasanna, Manish Vidyasagar

References

R.T.Q Chen. Isolating sources of disentanglement in variational autoencoders. In *NIPS*, 2018.

D.Kingma. Auto-encoding variational bayes. In *ICLR*, 2014.

D.Kingma. Deep visual analogy-making. In *NIPS*, 2015.

E.Nalisnick. Stick-breaking variational autoencoders. In *ICLR*, 2017.

Higgins. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

H.Kazemi. Style and content disentanglement in generative adversarial networks. In *IEEE WACV*, 2019.

H.Kim. Deep visual analogy-making. In *ICML*, 2018.

I.Sutskever. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

L. Dinh K. Goel A. Courville Y. Bengio J. Chung, K. Kastner. A recurrent latent variable model for sequential data. In *NIPS*, 2015.

Mathieu. Disentangling factors of variation in deep representation using adversarial training. In *NIPS*, 2016.

W.Joo. Dirichlet variational autoencoders. In *ICLR*, 2019.

Li Y. Disentangled sequential autoencoder. In *ICML*, 2018.

Appendix

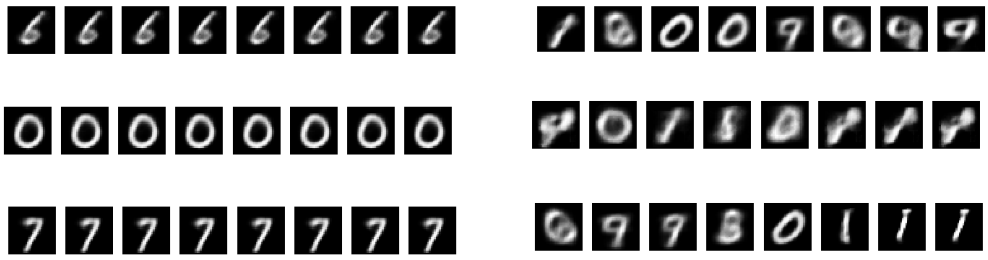

Figure 5: Some of the failure cases generated by the model.(Left) Fixed content and varying style and (Right) Fixed style and varying content. These images give further hint that content is not being learned accurately, and is either being passed on in the style variable or is being transmitted by the strong prior of VRNN which gets passed every timestep.