VQVAE FOR SPEECH PROCESSING

Sai Krishna Rallabandi and Alan W Black
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{srallaba, awb}@cs.cmu.edu

Abstract

This is a writeup for Speech Lunch. The idea is to provide intuitions of (hopefully correct), cover reasoning behind and document the implementation details of VQVAE. Specifically, we are concerned with the applications facing speech processing.

Index Terms: disentanglement, latent representation, continuous, discrete

1 Introduction

Blurb - We need to bring technology closer to humans
In order to bring technology closer to humans it is important to provide mechanisms for them to interact with the same. For instance, language technologies such as Speech Recognition and Synthesis, Visual Question Answering to name a few have been shown to be massively useful in this context. Having said that, there are three major bottlenecks for realization of this beautiful goal of all pervasive AI: (1) These technologies are currently only accessible in a handful number of languages around the planet. In order to have a meaningful impact it is imperative that such technologies need to at least exist in many more languages. (2) These technologies work for most people most of the time but not for all the people all of the time. In other words, these technologies lack the last mile reach and they fail in some scenarios. Failure is ok when doing speech recogition or playing chess or things of that nature, but not failing is critical in some scenarios. Or failing so scantily that it is less than negligible. Take autonomous vehicles for example. We want to have machines that fail once every 100 life times or a thousand lifetimes. That is when humans can sleep in an autonomous car. Thats when it is good enough to pass the regulations in terms of safety. (3) Worse, some of these models are akin to black boxes and we cannot even interpret what is happening inside these models. To realize the goal though this is an important step because the machine performance has to be under some interpretibility for everyday life usage. Take Google Trump thing as example. However, building such technologies is expensive in terms of annotated data, etc. On the other hand, social media and web 2.0 has enabled an outburst of audiovisual content at an unprecedented rate. Therefore, it might be useful to design techniques that can leverage such resources to accelerate the process of building a language technologies stack in under resourced languages.

Blurb - Lexicon is a good place to start with
A major bottleneck in the progress of many data-intensive language processing tasks such as speech recognition and synthesis is scalability to new languages and domains. Building such technologies for unwritten or under-resourced languages is often not feasible due to lack of annotated data or other expensive resources. A fundamental resource required to build such a stack is a phonetic lexicon - something that translates acoustic input to textual representation. Having such a lexicon, even if noisy, can help bootstrap speech recognition models, synthesis, and other technologies.

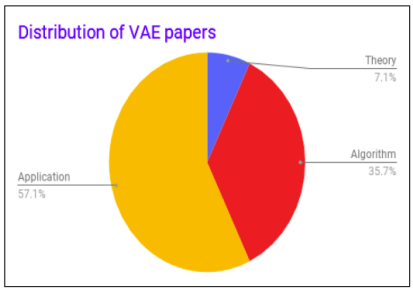
32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

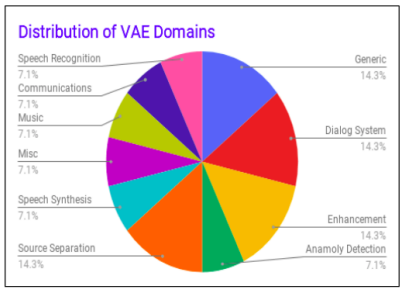
Design Choice - Resynthesis is a good proxy and Neural Generative Models are good
Artificial generation of speech based on neural approaches has soared in the recent past. There have been continuous and significant improvements in both the aspects of speech generation - fidelity and flexibility. These models aim to model the joint probability of the data distribution and the conditioning information as a product of conditional distributions. Autoregressive models such as (14), flow based models such as (8) have shown to generate audio that rivals the quality of natural speech. Approaches such as (13; 12) have shown ways to incorporate inductive biases into the generative process. (16) developed generic methods to enable the usage of distributional analysis of text at phone, word, and character levels in an unsupervised fashion. These techniques have been utilized in building highly flexible systems capable of generating different styles of speech and ability to build voices from noisy or very minimal data.

Design Choice - Latent Stochastic Variable Models
Speech has a lot of natural variations in terms of content, speaker, channel information, speaking style, prosodic variations, etc. Accordingly, we are interested in models which have flexibility to marginalize such variations but preserve the phonetic content and distinguish meaningful differences between phonetic units. To accomplish this, we employ sequence to sequence models with latent random variables (referred to as latent stochastic models hereafter). These models provide a mechanism to jointly train both the latent representations as well as the downstream inference network. They are expected to both discover and disentangle causal factors of variation present in the distribution of original data, so as to generalize at inference time. While training latent stochastic models, optimizing the exact log likelihood can be intractable. To address this, a recognition network is employed to approximate the posterior probability using reparameterization (6). When deployed in encoder-decoder models, this approach is often subject to an optimization challenge referred to as KL-collapse (1), wherein the generator (usually an RNN) marginalizes the learnt latent representation. Typical approaches to dealing this issue involve annealing the KL divergence loss (1; 18), weakening the generator (17) and ensuring the recall using bag of words loss.

1.1 Context: ICASSP 2019

There are 25 papers that mention the word 'Variational' in title.





Variational AutoEncoders come under a class of models that can be termed as latent stochastic variable models - they employ latent random variables in their framework. There are numerous references about different components of a Variational AutoEncoder and its interpretations.

1.2 Why so popular?

Latent Stochastic variable models provide a mechanism to jointly train both the latent representations as well as the downstream inference network In other words, they are expected to both discover and disentangle causal factors of variation present in the distribution of original data, so as to generalize at inference time. This ability to incorporate random variables within their framework and the potential to identify causal factors of variation in input data makes variational models attractive with respect to many applications from feature extraction through data augmentation. These applications cover both generative processes - in terms of the ability of such models to generate novel content - to discriminative processes - in terms of the robustness of such models to noise and other perturbations. There are works that combine these advantages and generate additional data for a discriminative task as well.

2 Why should this VQVAE work at all?

In this section, we first present our analysis of the optimization that happens in WaveNet, followed by latent stochastic models. We then present a case for controlling the disentanglement.

2.1 Analysis of optimization and disentanglement in Stochastic Generative Models for Speech

Lets take WaveNet. WaveNet (15) is an autoregressive neural model with a stack of 1D convolutional layers that is capable of directly generating audio signal. It has been shown to produce generated speech that rivals natural speech when conditioned on predicted mel spectrum (10). The input to WaveNet is subjected to corresponding gated activations while passing through each dilated convolutional layer and is classified by the final softmax layer into a μ law encoding. The concrete form of the residual gated activation function is given by following equation:

$$r_d(x) = tanh(W_f * x) \odot \sigma(W_g * x) \tag{1}$$

where x and $r_d(x)$ are the input and output with dilation d, respectively. The symbol $*$ is a convolution operator with dilation d and the symbol $\odot$ is an element-wise product operator. W represents a convolution weight. The subscripts f and g represent a filter and a gate, respectively. The joint probability of a waveform $\mathbf{X}$ can be written as:

$$P(X|\theta) = \prod_{t=1}^{T} P(x_t|x_1, x_2..x_{t-1}, \theta) \tag{2}$$

given model parameters θ. During implementation of WaveNet, the autoregressive process is realized by a stack of dilated convolutions. The final output y_t at time step t can be expressed mathematically as:

$$\hat{y_t} \sim \sum_{d=0}^{D} h_d * r_d(x) \tag{3}$$

where x, y represent input and output vectors; D is the number of different dilation used and d is the dilation factor; h_d is the convolution weights. This stack of convolutions is repeated multiple times in the original WaveNet. Optimization in WaveNet is performed based on the error between predicted sample and the ground truth sample conditioned on previous samples in the receptive field alongside the local conditioning. Expressing the loss function being optimized mathematically the error at sample t is:

$$l_t = Div(\hat{y_t}||y_t) \tag{4}$$

Here, we define the divergence similar to the (9), To optimize this loss, the contribution from the individual convolution layers towards this global error function must be nullified. Now let us consider the expression for intermediate output for a single filter in Eqn 3:

$$x_{out}(t) = \sum_{\tau=0}^{t} h(\tau)x(t-\tau) \tag{5}$$

where τ is the receptive field covered by the model and $h(\tau)$ represents the discrete state representation at time t. Without loss of generality and dropping the term τ for brevity, the spectral representation generated by the model can be expressed as:

$$Y(z) = H(z)X(z) \tag{6}$$

Considering the discrete nature of input from Eqn 4, an interpretation of Eqn 6 is that the neural autoregressive model acts as the transfer function and is discretized by convolving with the samples

from original signal. It has to be noted that this is similar to the formulation of source filter model of speech, specifically the periodic components aka voiced sounds. Voiced sounds typically represented as impulse train are convolved with the transfer function to generate spectral envelope. As a corollary, from Eqn 4 and 6, we posit that the optimization in WaveNet model is performed by minimizing the divergence between true and approximate spectral envelope. Note that latent stochastic models such as VAEs are aimed to minimize the divergence between true and approximate posterior distributions of input data. The advantage with such models is the presence of stochastic random variables that capture the causal factors of variation in input based on some prior information about the distributional characteristics of data. Techniques aimed at this (5) have shown that it is possible to effectively disentangle the factors of variation using stochastic variables. Hence, we postulate that it should be possible to augment WaveNet decoder with a suitable encoder and an appropriate prior distribution to disentangle the acoustic phonetic units from a given utterance.

2.2 Analysis of role of priors in Latent Stochastic Models: Interplay between disentanglement and reconstruction

Let us consider the ELBO being optimized by Variational AutoEncoders:

$$E_{q_\phi(z|x,c)}[logp_\theta(x|c,z)] - |D_{KL}(q_\phi(z|x,c)||p_\theta(z|c))| \tag{7}$$

where the first term is the reconstruction error while the second is the divergence between approximate and true posteriors. The KL term plays in important role in the optimization by forcing the posterior distribution output by encoder to follow an appropriate prior about the data generation process. The global optimum value for this term is therefore 0, and is reached only when both the distributions exactly match each other. Since the prior information about the data generation process typically involves some causal factors of variation of the data, this naturally translates to a requirement on the encoder to track such factors. It has to be noted that during training optimization is performed in expectation over minibatches. The Expectation of KL term can then be rewritten as related to the amount of mutual information between the latent representation and the data distribution (7). Therefore, as the KL term decreases, so does the amount of information the encoder can place in the latent space. In order to facilitate the reconstruction with this limited capacity, the encoder has to also discard some nuisance factors that may not have contributed to the generation of data. Thus, the KL term forces the encoder network to disentangle the causal factors - akin to the basis vectors - of variation in the data and ignore the nuisance factors that may have contributed to the generation of data.

However, if the prior is too simplistic such a unit normal distribution, the model is trivially incentivized to force the posterior distribution to closely follow the Gaussian prior distribution (3) especially early during the training. Typically the decoders in variational models are implemented using powerful universal approximators such as RNNs. In the context of speech, a typical example of a generative model that can be applied as the decoder is an autoregressive framework such as WaveNet. Since these decoders are very powerful, they are able to learn the priors about data distribution by themselves. When this happens, they ignore the latent representation input from the encoder and the prediction of next sample is based solely on the marginal distribution at the current timestep by using a dictionary approach. Therefore, the encoder is no longer forced to track the causal factors of variation in the data, leading to issues such as mode collapse.

A reasonable and intuitive solution therefore is to make the prior space more complex thereby pressurizing the posterior distribution to track the prior space more closely. For instance, (2) attempt to accomplish this by adding a hyperparameter β to promote disentanglement and gradually increasing channel capacity, something that increases loss. This can be expressed as follows:

$$E_{q_\phi(z|x,c)}[logp_\theta(x|c,z)] - \beta|D_{KL}(q_\phi(z|x,c)||p_\theta(z|c)) - C_z| \tag{8}$$

where C_z is the channel capacity term (2). However, it has to be noted that simply making the prior distribution arbitrarily complex also perhaps leads to unreasonable constraints on the decoder. For instance, consider tasks in Natural language processing such as language modeling, machine translation, image captioning. These are known to be characterized by heavy tailed Pareto output distributions. In such scenarios, it is unreasonable to expect a uniform Gaussian prior distribution for

the latent generative process. Employing such priors leads to weakening the decoder and resulting in poor reconstruction ability. Therefore, priors in latent stochastic models play a significant role in the optimization. They facilitate disentanglement of causal factors of variation on the one hand, as well as control the ability of the model to reconstruct the data distribution on the other.

2.3 Case for Controlled Disentanglement

We believe that complete disentanglement of input data into its independent causal factors of variation is not fully useful. A more attractive option is to employ priors about the causal factors of variation and control what gets disentangled. This can also be seen as a way of incorporating inductive bias into the model. Inductive biases are good because they help with interpretibility. Not having priors might in fact lead to some clusters that do not conform to any logical explanations. While this might be good initially, we need interpretibility when we want to use these models as first class objects to build a stack of technology. This was always the case in supervised learning anyway. The idea is to do away with the nuisance variables. Typical priors used today however, are generic and include every factor of variation. This is not only less useful but also leads to an average performance. A manifestation of this can be seen in the spoilt reconstruction ability of the models that use global priors. Instead of weakening the decoder for covering the reconstruction, we can flip the equation and force the encoder to encode only information required by the decoder, employing the prior to accurately solve the task at hand.

Let us consider a data distribution X which consists of class examples $\{x_1, x_2..., x_n\}$, where each x_i is described by attribute-set (a, b, c). The prior distribution of X can be represented by a parameteric function g such that g maximizes the likelihood of X over the set of its attributes:

$$P_\omega(X) = g_\omega(a, b, c) \tag{9}$$

Note that the attribute-set can either contain individual entities or the relationships between them or both. To illustrate this, let us consider a toy-example where we build a binary classifier to predict if a given integer triplet is a Pythagorean triplet. Pythagorean triplets are a triplet of numbers that follow Pythagoras Theorem such as $\{3, 4, 5\}$ and $\{5, 12, 13\}$. In this task, the attribute-set consists of the relationship between the first two-elements of the triplet. If the model is able to discover this attribute, it can generalize for any given numbers. However, if we have a more complicated task like building a classifier for MNIST digits, then the attribute-set has multiple first and second order relations like brush strokes, shape of the digits etc. The success of modelling $P_\omega(X)$, and ultimately the success on the downstream task, relies on how well can the model disentangle these individual attributes from the observed data X_t. Mathematically, let us consider the posterior probability of a training instance x_1 expressed as

$$P_\theta(x_1) = f_\theta(x_1) \tag{10}$$

where *f* denotes arbitrary function and θ denotes the parametric family used to model the distribution X. It can be seen that compositionality over an unseen training instant x_{new} would be possible if *f* is related to *g*. In other words, *f* needs to have some information about the latent causal factors of variation that generated X in the first place. We hypothesize that this can be accomplished by using a model that has the potential to generate the attribute-set in this context. Once this is done, the test instance can be appropriately expressed as

$$P_\theta(x_{new}) = h(a, k(b, c)) \tag{11}$$

where h and k can be a novel combination of functions that embed these attributes in the manifold of original distribution of X.

3 Implementation

The architecture of our model is built on top of VQ-VAE. It consists of three modules: an encoder, quantizer and a decoder. As our encoder, we use a dilated convolution stack of layers which downsamples the input audio by 64. The speech signal was power normalized and squashed to

the range (-1,1) before feeding to the downsampling encoder. To make the training faster, we have used chunks of 2000 time steps. This means we get 31 timesteps at the output of the encoder. The quantizer acts as a bottleneck and performs vector quantization to generate the appropriate code from a parameterized codebook. We define the latent space $e \in R^{k \times d}$ to contain k d-dim continuous vector. Quantization is implemented using minimum distance in the embedding space. The number of classes was chosen to be 64, approximating 64 universal phonemes. Assuming $z_e(x)$ denotes the encoder output in the latent space, then the input of decoder $z_d(x)$ will be obtained by ${}_jd(e_j, z_e(x))$, where d is a similarity function of two vectors. In this paper, we consider Euclidean distance as the similarity metric. Our decoder is an iterated dilated convolution-based WaveNet that uses a 256-level quantized raw signal as the input and the output from vector quantization module as the conditioning. Although using a Mixture of Logistics loss function might yield a better output, we have only used a 256 class softmax in this study. The decoder takes the output from the quantizer along with the speaker label as global conditioning and aims to reconstruct the input in an autoregressive fashion. Following IDCNNs, we have shared the parameters of all the stacks. Although a structure such as HMM (4) might be intuitively better at capturing the transitions, we have limited ourselves to a vector quantization based approach as we observed it to be better at handling the posterior collapse problem in VAEs. We have used 3 stacks of 10 layers each in the decoder and residual blocks with similar dilation factors in each of the stacks shared their parameters (11).

References

[1] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[2] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[3] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

[4] J. Ebbers, J. Heymann, L. Drude, T. Glarner, R. Haeb-Umbach, and B. Raj. Hidden markov model variational autoencoder for acoustic unit discovery. In *INTERSPEECH*, pages 488–492, 2017.

[5] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[6] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[7] A. Makhzani and B. J. Frey. Pixelgan autoencoders. In *Advances in Neural Information Processing Systems*, pages 1975–1985, 2017.

[8] R. Prenger, R. Valle, and B. Catanzaro. Waveglow: A flow-based generative network for speech synthesis. *arXiv preprint arXiv:1811.00002*, 2018.

[9] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[10] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *arXiv preprint arXiv:1712.05884*, 2017.

[11] E. Strubell et al. Fast and accurate entity recognition with iterated dilated convolutions. 2017.

[12] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani. Voice synthesis for in-the-wild speakers via a phonological loop. *arXiv preprint arXiv:1707.06588*, pages 1–11, 2017.

[13] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani. Voiceloop: Voice fitting and synthesis via a phonological loop. *arXiv preprint arXiv:1707.06588*, 2017.

[14] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*, 2016.

[15] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[16] O. Watts. *Unsupervised Learning for Text-to-Speech Synthesis*. PhD thesis, University of Edinburgh, 2012.

[17] T. Zhao, R. Zhao, and M. Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*, 2017.

[18] C. Zhou and G. Neubig. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. *arXiv preprint arXiv:1704.01691*, 2017.