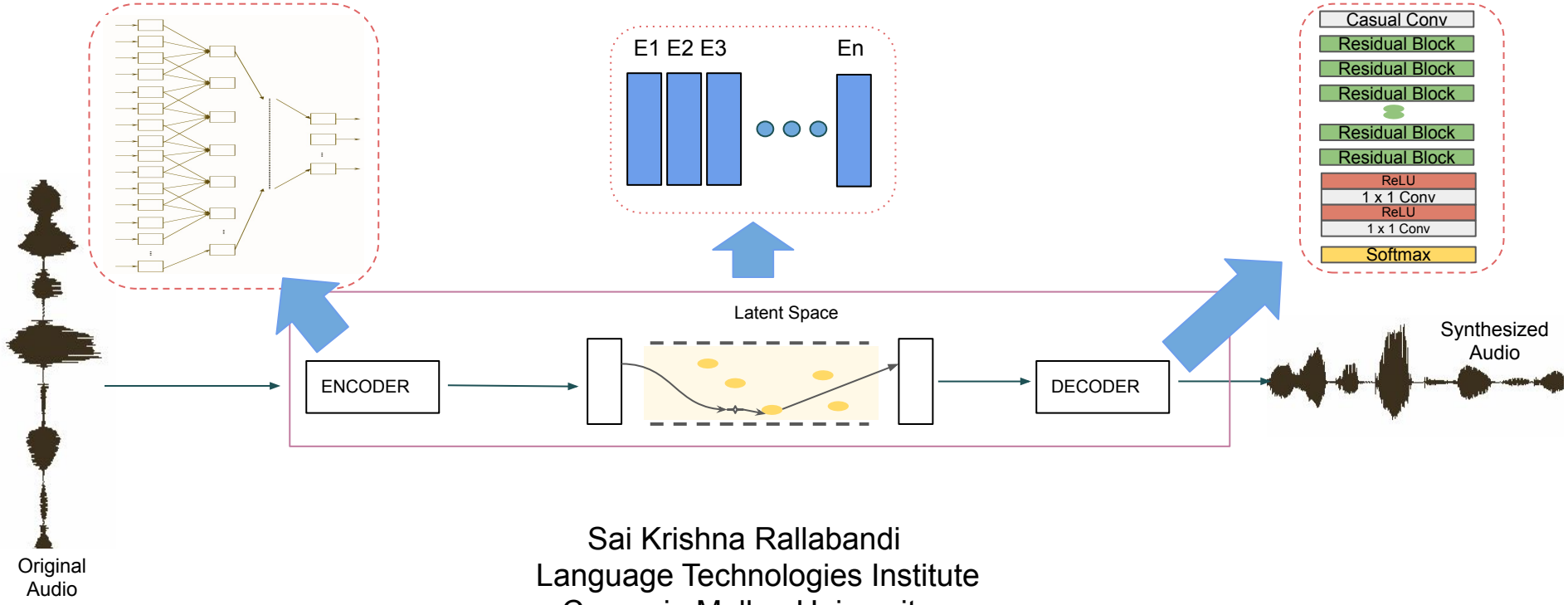
Latent Stochastic Variable Models for Speech: A Case study with VQ VAE



Sai Krishna Rallabandi
Language Technologies Institute
Carnegie Mellon University

Updates slides: http://www.cs.cmu.edu/~srallaba/pdfs/speechlunchpresentation_June2019_lsms.pdf

Agenda

Scope and Context : ICASSP 2019

Part 01: Optimization in Latent Stochastic Variable Models (~10 min)

- Overview of a typical deep learning architecture
- Marginalization of Nuisance Variables
- Disentanglement of Causal Factors of Variation in the input data

Part 02: Vector Quantized Variational AutoEncoder (~17 min)

- VQ in the latent space
- KLD as a hyperparameter
- Controlled Disentanglement using VQ VAE

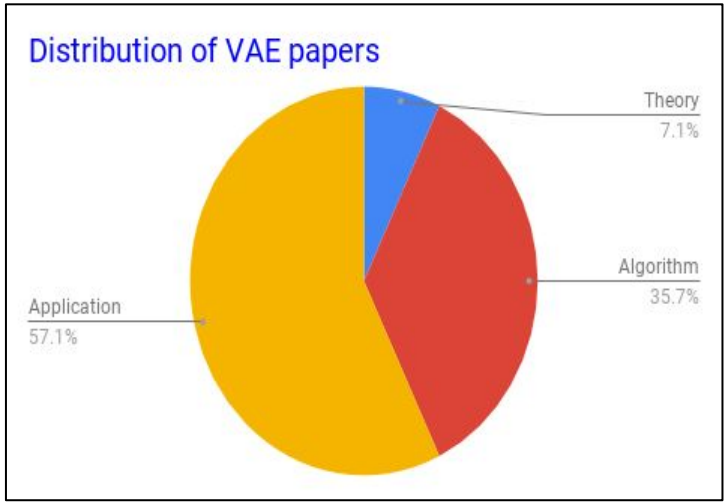
Part 03: VQ VAE for Speech Coding (~20 min)

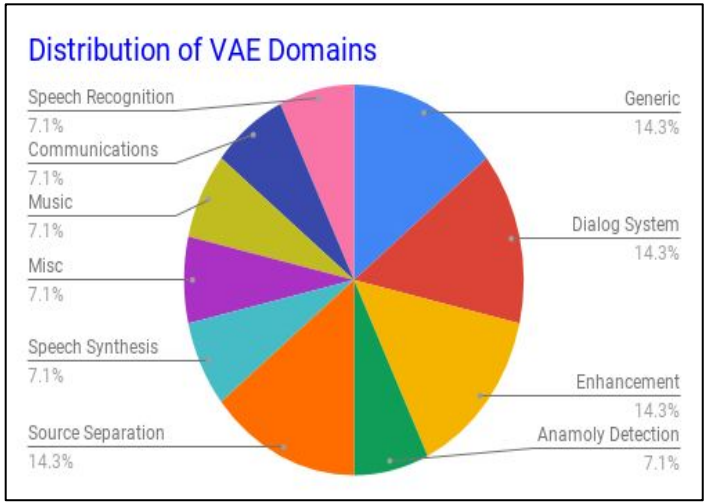
- Speaker Codebook
- F0 auxiliary loss

Conclusion

Scope and Context: ICASSP 2019

- Latent Stochastic Variable Models: Models that incorporate hidden random variables

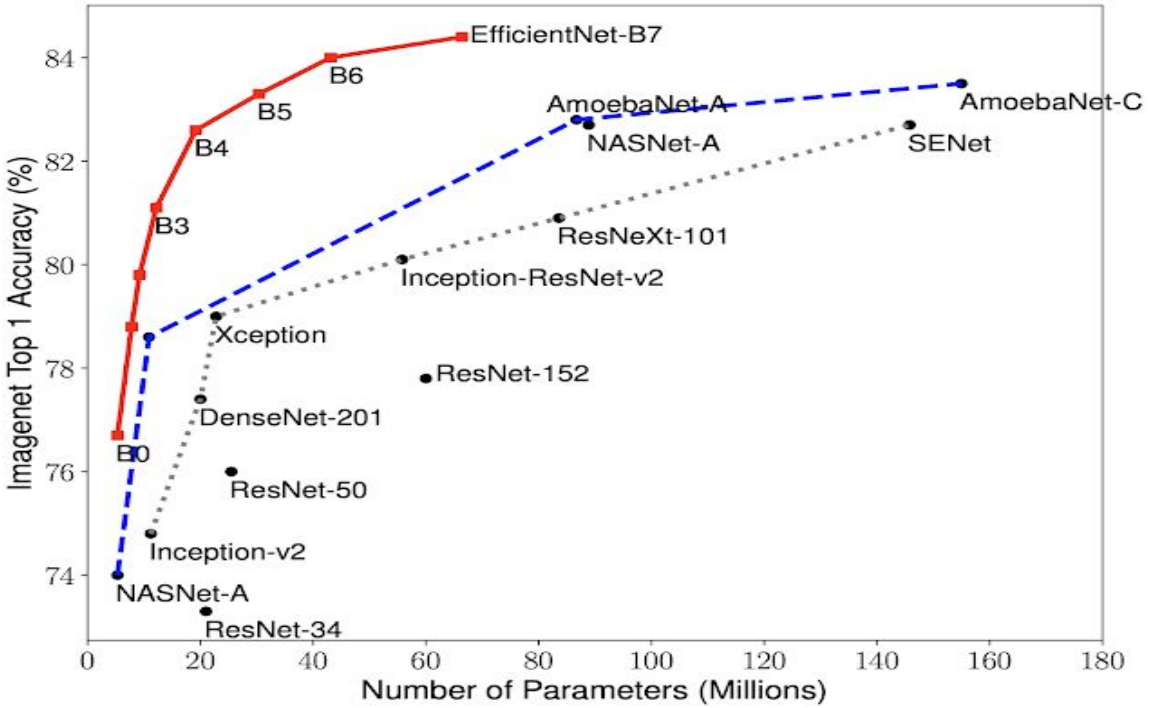




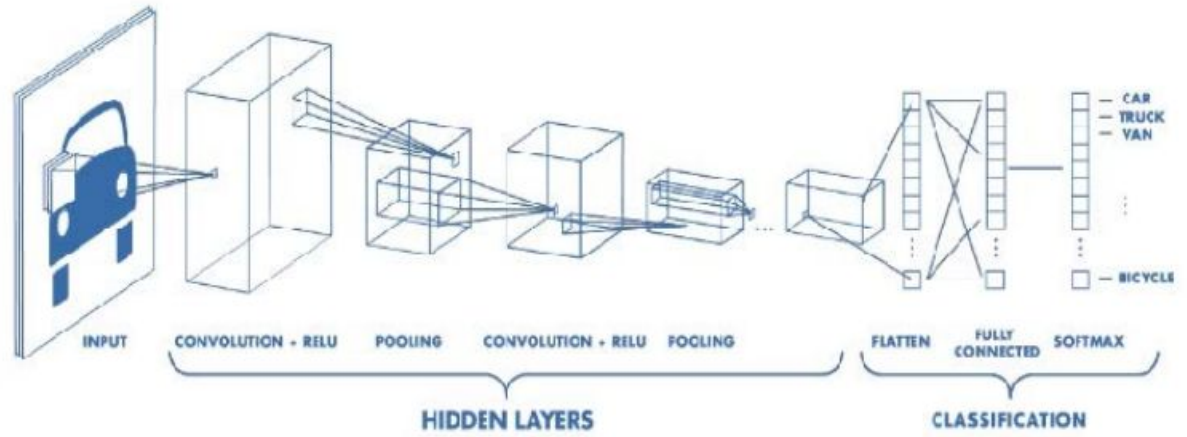
- ~2.5% papers that employ Variational Inference
- Generate novel samples in generative processes
- Robust to noise and attacks in discriminative processes

"*Latent Variable Models provide a declarative language for specifying prior knowledge and structural relationships in complex datasets*" - Alexander Rush, Tutorial, EMNLP 2018.

Part 01: Deep Learning Architectures



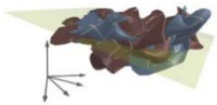
Part 01: Let's look under the hood of a deep learning architecture



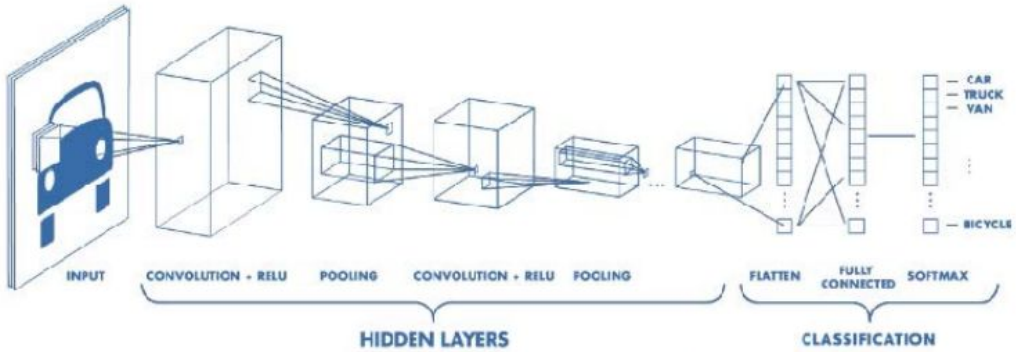
- Feature Extraction component
- Information Bottleneck component
- Classification component

"Deep Learning can be summarized as matrix multiplications followed by non linearity" - an abstractive oneliner from plenary talk by Prof. Hermann Ney, ICASSP 2019

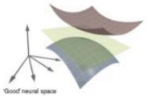
Part 01: Importance of Information Bottleneck



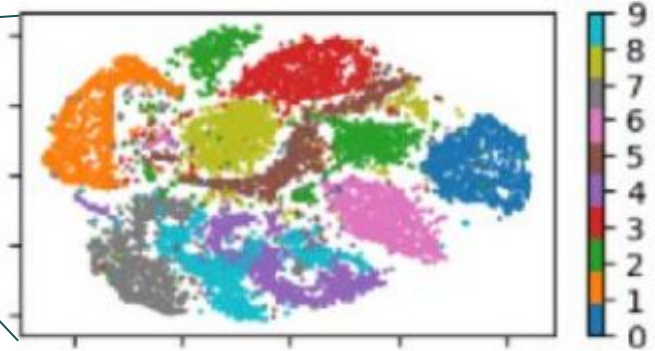
Raw feature space



- The model needs to discard nuisance factors [Marginalization]

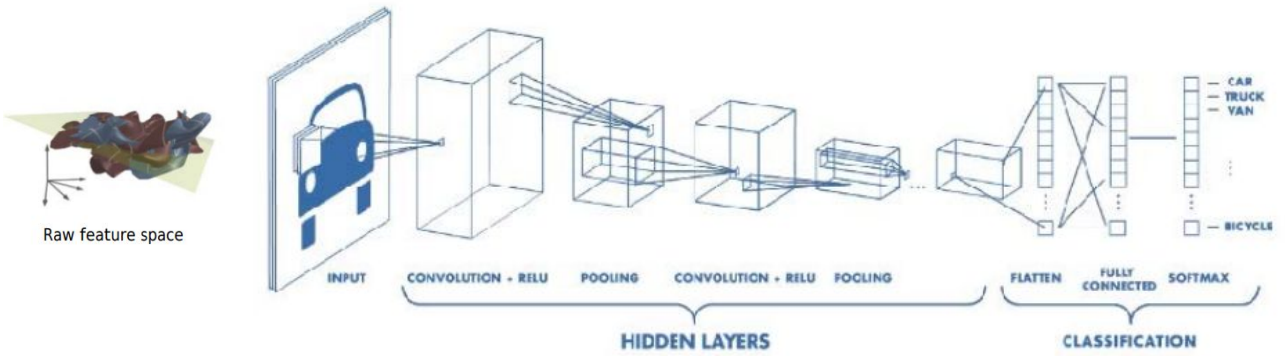


Latent space



"*The only thing that really matters is the mutual information between encoder and decoder*" - Prof Naftali Tishby, Data Science Summit 2017, Interspeech Tutorial 2018.

Part 01: Lets squeeze everything out of dimensionality reduction



Barebones Loss = Div (y_hat , y)

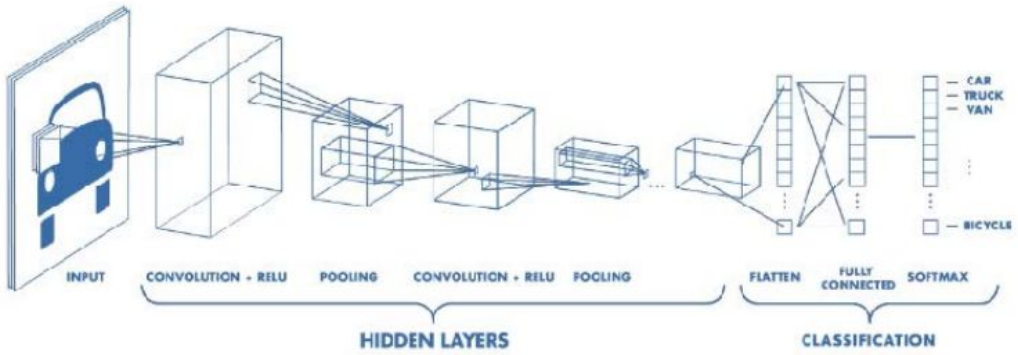
Regularized Loss = Div (y_hat , y) + ƛ (information bottleneck) where information bottleneck = div(x,z)

- Typically Div is realized by cross entropy and div by KL Divergence.

This is all but the equation of a VAE!

"Average Reconstruction and entropy are in tension with each other", ELBO Surgery, yet another way to carve up the variational evidence lower bound, NeurIPS 2016

Part 01: So what happens when we try to reconstruct the input space?



- AutoEncoding Loss = Reconstruction Loss + Regularized KL between input space and latents

- The model needs to capture all the causal factors of variation (distinct modes) present in the input distribution so that it can regenerate with high fidelity.
  [Disentanglement]

"We assume that the structure of the world dictates a certain natural decomposition", Towards a definition of Disentangled Representations, DeepMind, 2019

Part 01: Disentanglement vs Decomposition vs Factorization (?)

Part 01: Let's summarize the observations

The main characters in this story thus far:

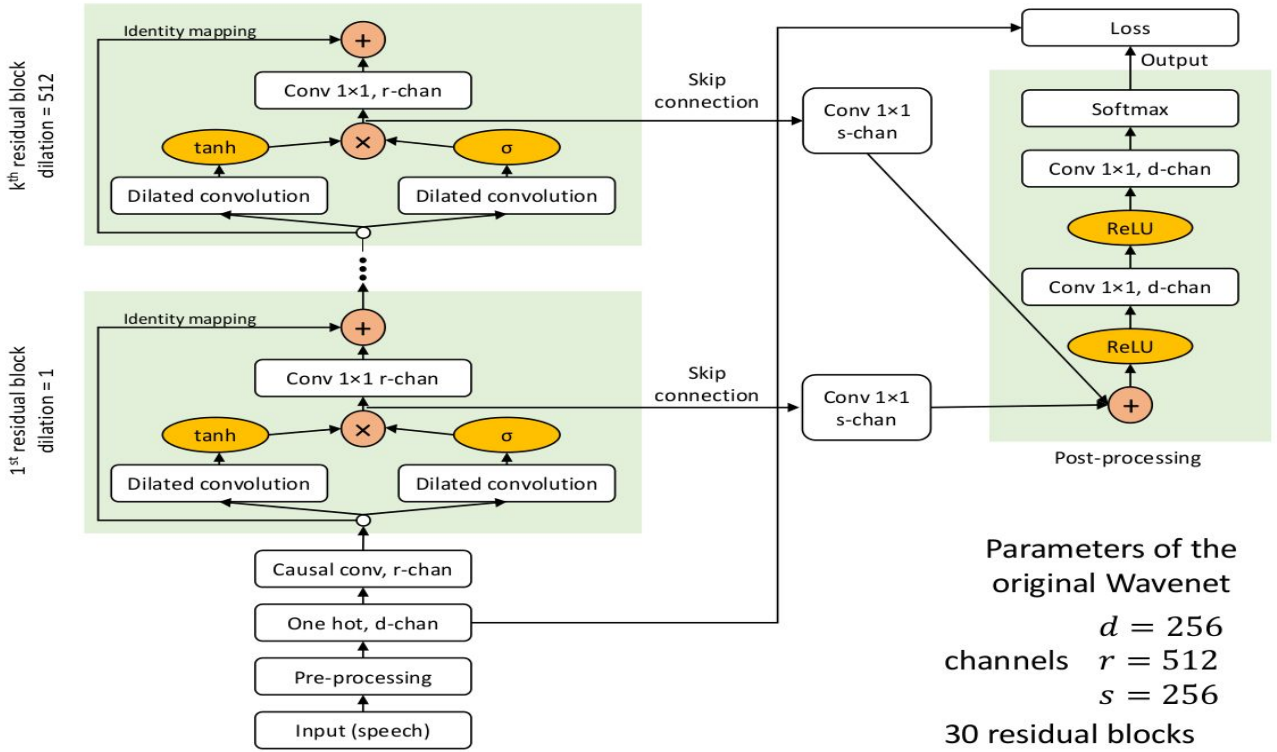
- Encoder: tries to marginalize the nuisance variables.
- Regularizer: tries to approximate optimal dimensionality reduction.
- Decoder: tries to reconstruct the input distribution.

Sidekicks:

- But KL Divergence is also the mutual information between input and latent space. In other words, log(KL) value is the amount of information encoder can place in the latent space.
- Conditioned decoders are powerful enough to generate y_hat by themselves.

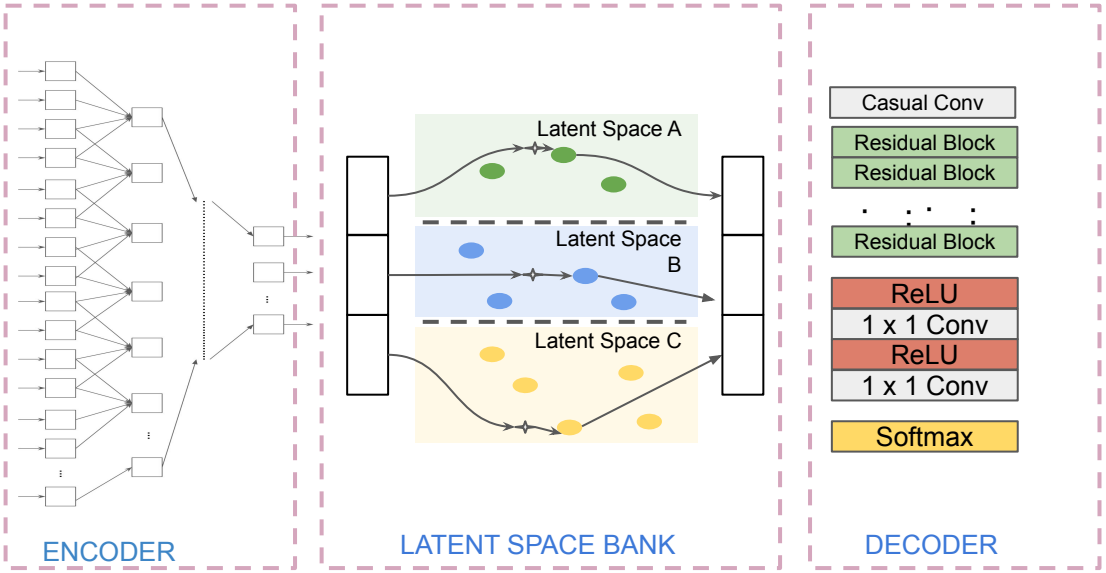
"On the one hand, it(Information Bottleneck) should provide the decoder with information necessary for perfect reconstruction and thus capture in the latents as much of the input data characteristics as possible. On the other hand, the constraints force some information to be discarded, preventing the latent representation from being trivial to invert, e.g. by exactly passing through the input", Unsupervised speech representation learning using WaveNet autoencoders, 2019

Part 02: What are our decoders? Let's consider WaveNet*

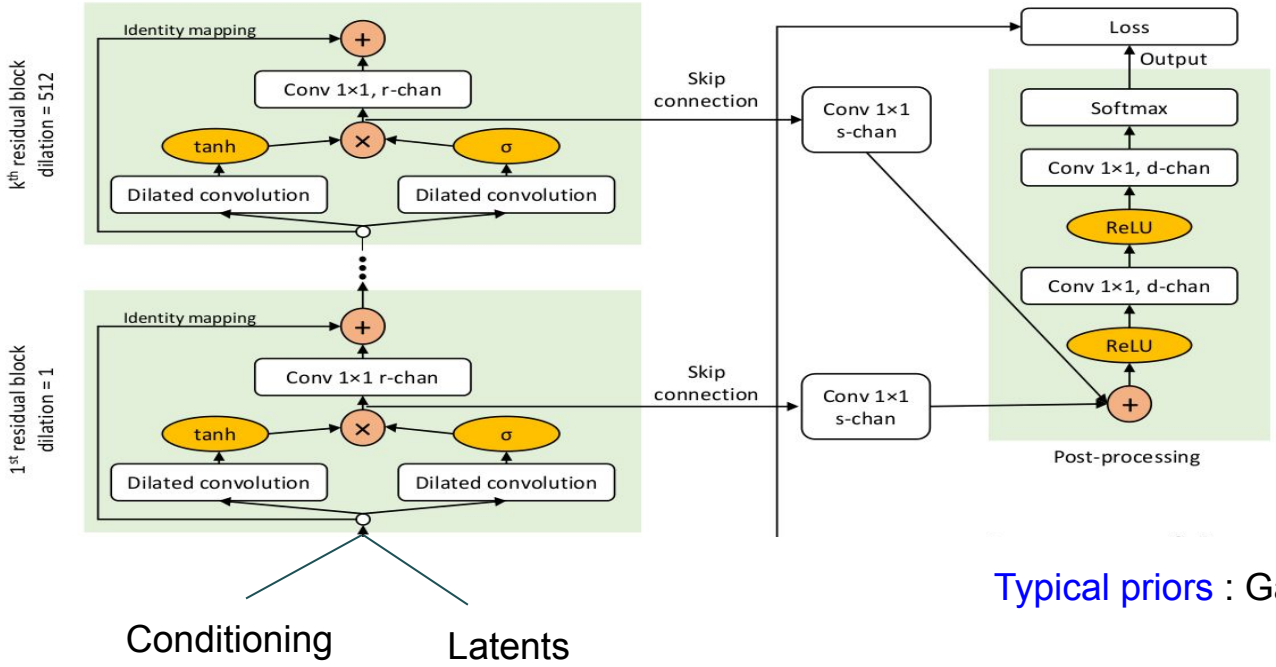


* http://www.cs.cmu.edu/~srallaba/ProjectPeregrine/reports/presentation_wavenet.pdf

Part 02: What happens to IB when a powerful decoder is employed?



Part 02: Let's look at this from Decoder

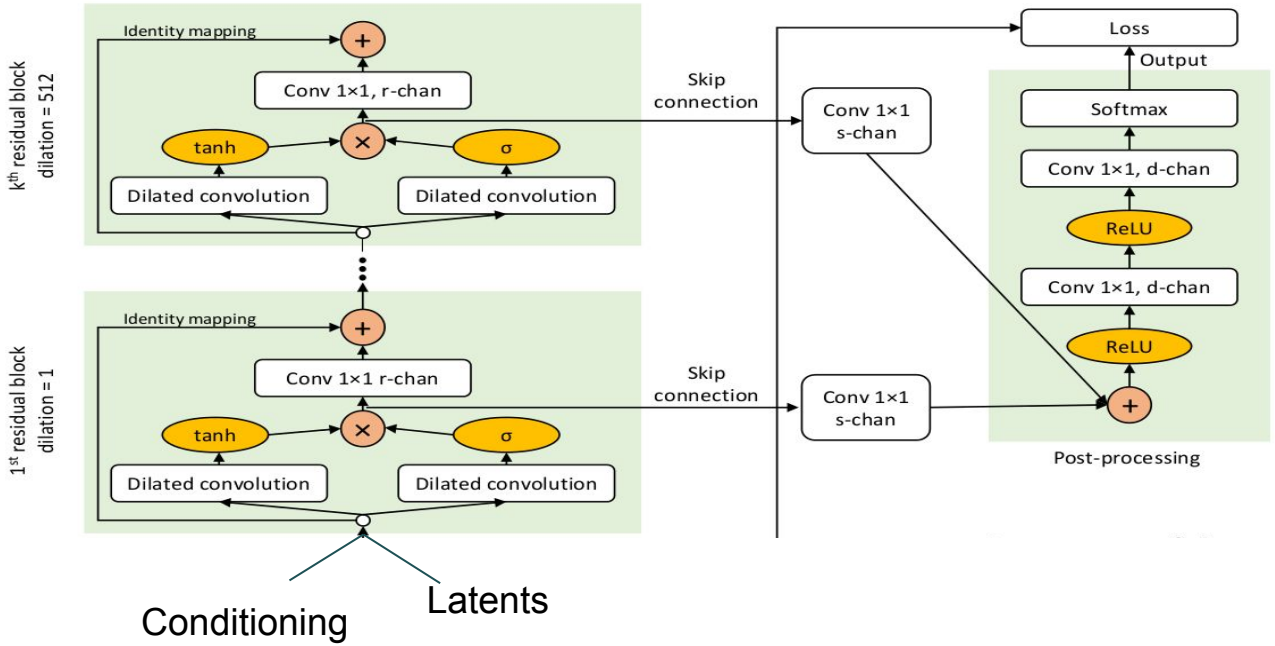


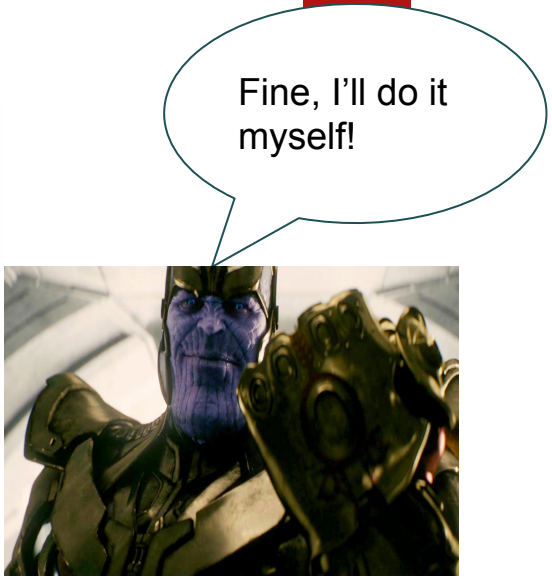
Typical priors : Gaussian

"Information that can be modeled locally by decoding distribution p(x|z) without access to z will be encoded locally and only the remainder will be encoded in z", Variational Lossy AutoEncoder ICLR 2017.

"The latent code z will be used when the two-part code is an efficient code", Variational Lossy AutoEncoder ICLR 2017.

Part 02: Decoders are a bit like…Thanos. As long as decoders are concerned, latents are extra dimensions that just aren't efficient.



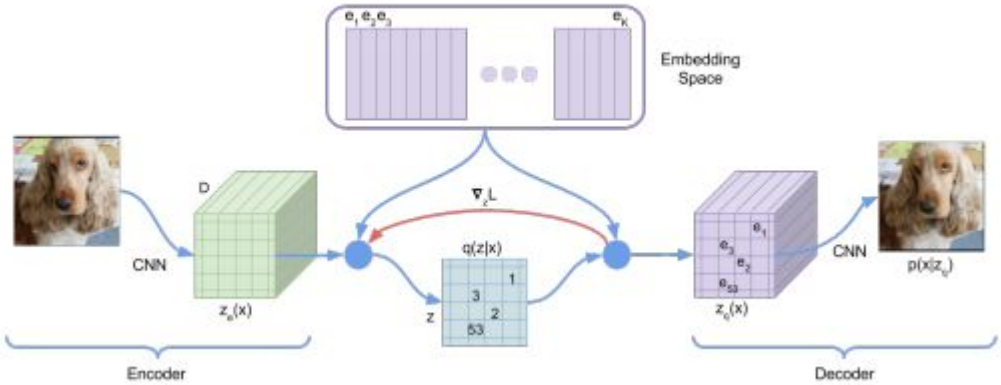


- Between decoders and encoders forcing marginalization, latent dimensions assume a unit normal distribution.

"The latent code z will be used when the two-part code is an efficient code", Variational Lossy AutoEncoder ICLR 2017.

Part 02: So how to train latent stochastic variable models reliably? VQ - VAE

- Autoregressive prior for the latents too +
- Using discrete latent representation instead of continuous +
- Clustering the latent space (learning the prior) = Vector Quantized VAE



Explanation of implementing Vector Quantization in the latent space: https://youtu.be/kS0kTZ2S150

Part 02: Speech Representation Learning by VQ - VAE

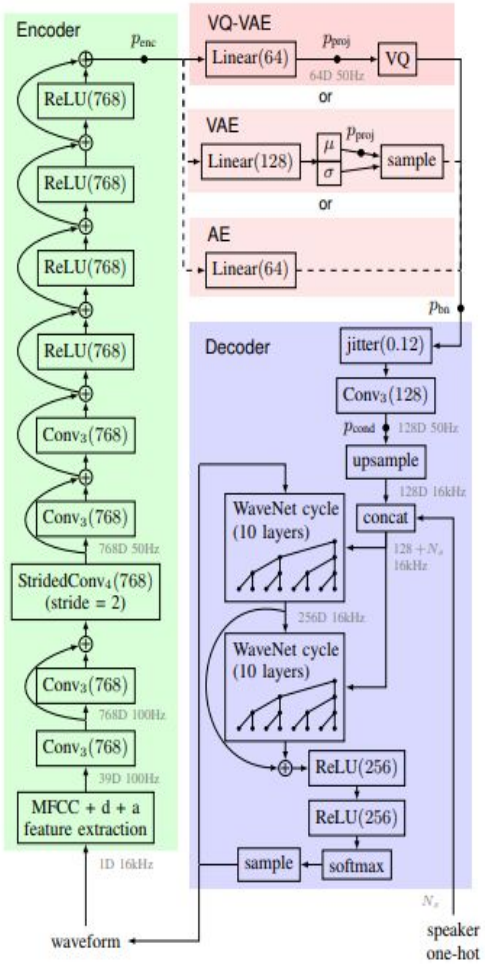
- to capture high level semantic content from the signal, e.g. phoneme identities, while being invariant to confounding low level details in the signal such as the underlying pitch contour or background noise.

- Decoder Loss: Reconstruction between prediction and target

- Embeddings Loss: Embeddings don't receive gradients from the decoder. Use L2 to move embeddings closer to encoder outputs. $|| sg[z(x)] - e ||^2$

- Commitment Loss: Encoder outputs need to be within a bound. Use L2 to move encoder outputs closer to embeddings. $|| z(x) - sg[e] ||^2$

- Decoder does not control the latent vectors anymore!



Link to the notebook with implementation of stop gradient in PyTorch: https://colab.research.google.com/drive/1rGzZtFzdwk9H2oWf2Eb3Ui0j9tJRkiSD

Part 02: Let's summarize the observations

- VQ VAE employs discrete latent variables and performs clustering in the latent space
- No Posterior collapse
- Can be used to model speech
- Can be used to extract unsupervised representations from raw audio

"Information that can be modeled locally by decoding distribution p(x|z) without access to z will be encoded locally and only the remainder will be encoded in z", Variational Lossy AutoEncoder ICLR 2017.

Part 03: Speech Coding by VQ - VAE

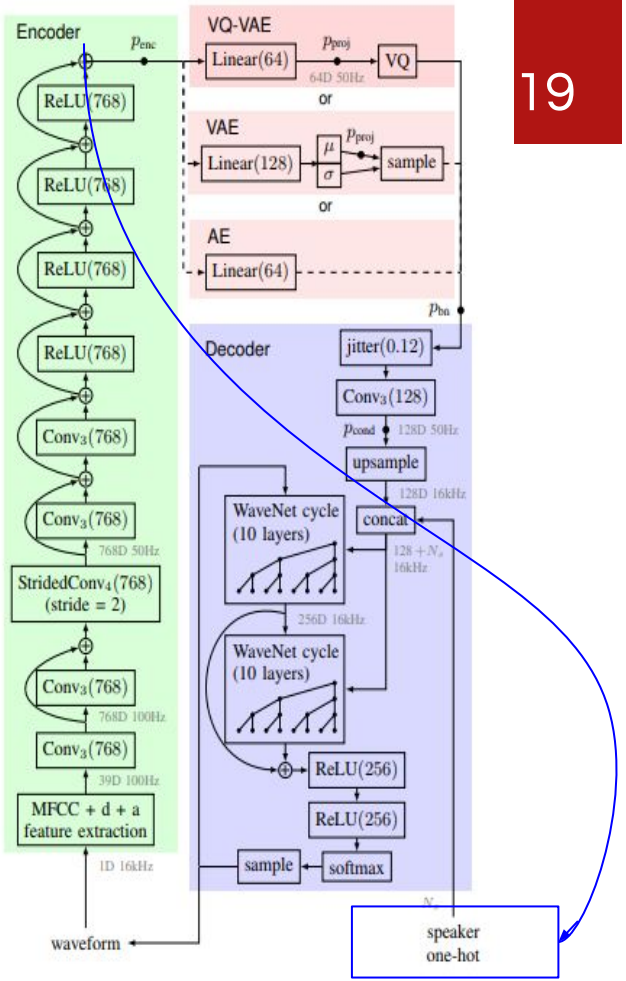
- How to efficiently store/transmit speech? - minimally redundant representation
- Speech codecs use hand engineered encoder and decoders
- High quality speech codecs > 16 kbps
- Learn redundancies in speech and exploit them
- True information rate of speech < 100 bps*
- Neural architectures learn redundancies by being exposed to multiple examples during training.

**"We decomposed the spectral representation that minimizes the bottleneck into eigenvectors that span time and frequency by stacking successive spectra"* , ON THE INFORMATION RATE OF SPEECH COMMUNICATION, ICASSP 2017.

Part 03: Changes to VQ VAE Architecture

SPEAKER IDENTITY

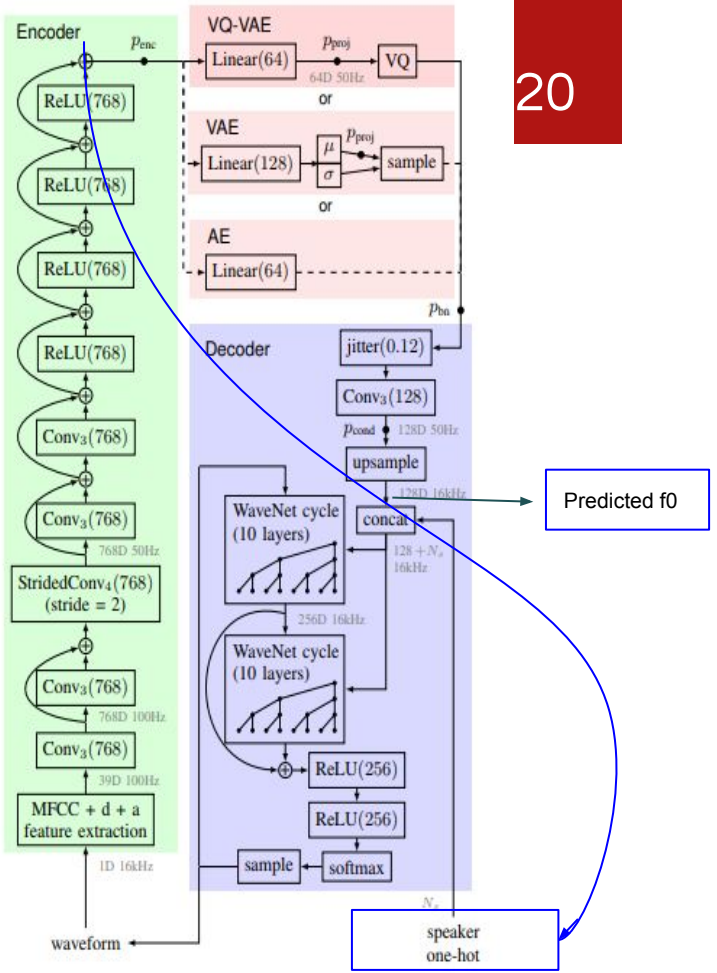
- Learnt Speaker Identity instead of explicit
- Mean pooling over the time dimension of the encoder
- Fed to a separate code book
- Time varying codes - message
- Time invariant codes - speaker
- Over entire utterance at test time



Part 03: Changes to VQ VAE Architecture

CONTROLLING PROSODY

- Additional loss to predict f0
- After upsampling layer
- *produce an utterance that has the original pitch track*
- f0 prediction module ignored when trained as codec
- f0 was extracted at 200 Hz
- Extra convolution layer at encoder for rate compatibility.



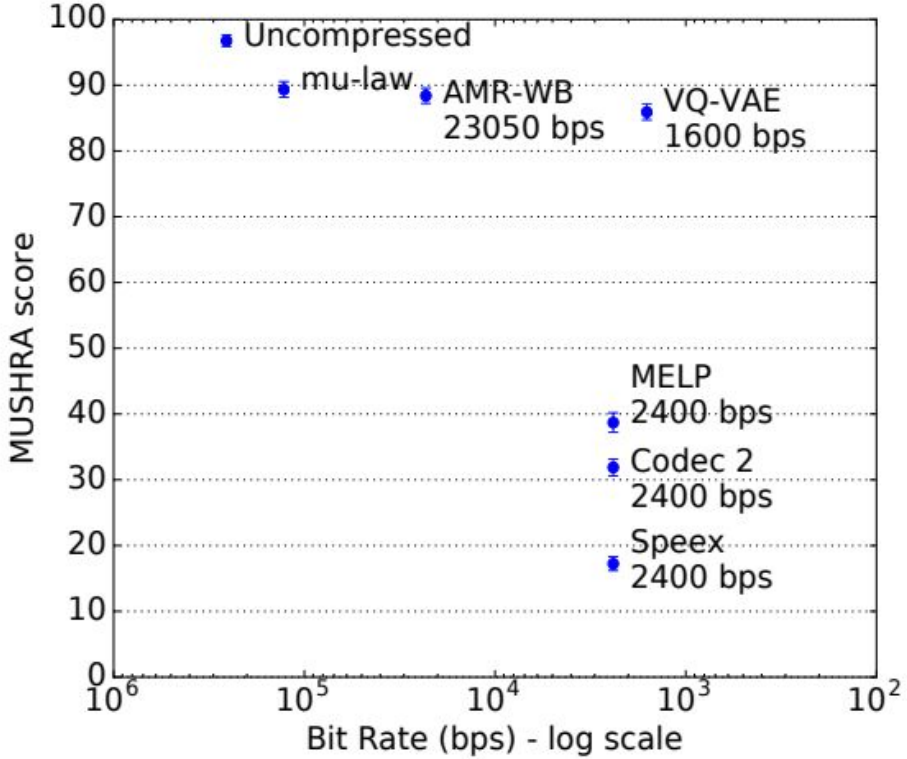
Part 03: Evaluation

Datasets:

- LibriSpeech (1000 hours, 2302 speakers): Combined train-clean-100, train-clean-360, train-other-500 for training ; test-clean for evaluation
- Augmented LibriSpeech(LSPlus): 0.01 of test-clean for eval and rest added to train
- Studio Corpus: 10 American speakers 300 hours - 85% recorded in studio environments

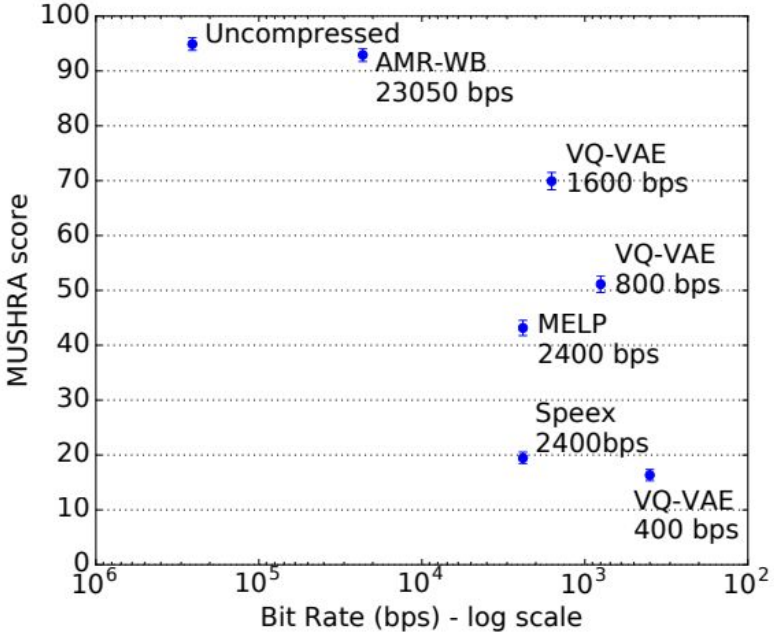
Metric: MUSHRA

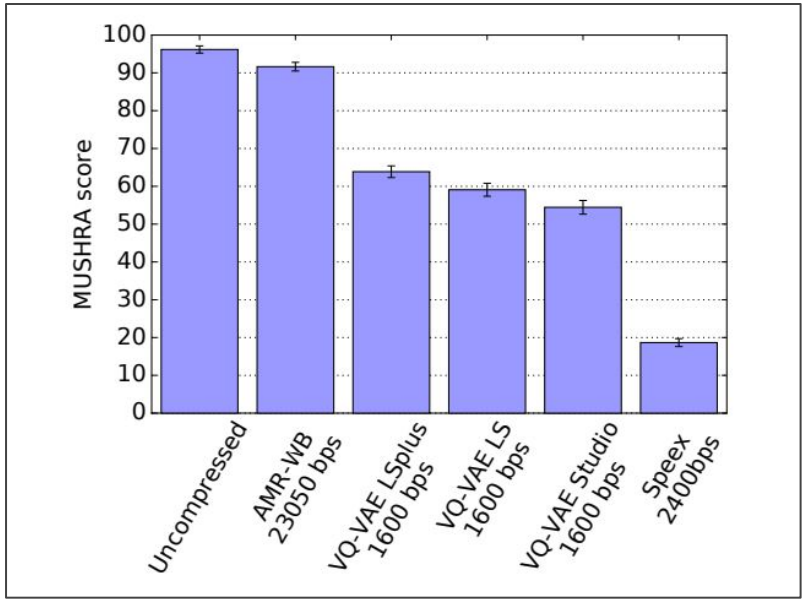
MUSHRA score vs bit-rate for Studio Corpus at 1.6 kbps, Bit rate reduces from left to right



- Two clusters emerge
- 14x improvement

MUSHRA score vs bit-rate for LibriSpeech Corpus, Bit rate reduces from left to right





- Even at 800 bps, the VQ-VAE model outperforms MELP at 3 times the bitrate, and it is only at 400 bps that performance degrades below that of Speex at 2400 bps

Part 03: Evaluation

Speaker Similarity Test

Codec	Mean Opinion Score
VQ -VAE LSPlus 1600 bps	3.7 +/- 0.451
VQ -VAE LS 1600 bps	3.703 +/- 0.716
MELP 2400 bps	3.138 +/- 0.324
Speex 2400 bps	2.534 +/- 0.233

Observations

- Lowest MoS Score for one speaker 1.85. Next lowest 3.38
- MoS when this speaker was included in the training set: 2.73

Conclusions

- VQ VAE at 1600 bps outperforms some popular low rate codes.
- More work required to understand variability in speaker similarity.
- Model is expected to achieve compression beyond 1600 bps.

Some useful links

- Writeup accompanying talk: http://www.cs.cmu.edu/~srallaba/ProjectPeregrine/reports/speechlunch_June2019_vqvae.pdf
- Samples from VQ VAE: http://tts.speech.cs.cmu.edu/rsk/challenges/zerospeech2019/kitchens/kitchen_vqvrnn/model_outputs/vqvae.43.upconv/
- Explanation of Vector Quantization in the latent space: https://youtu.be/kS0kTZ2S150
- Link to the notebook with implementation of core components: https://colab.research.google.com/drive/1rGzZtFzdwk9H2oWf2Eb3Ui0j9tJRkiSD
- Docker image with implementation: srallaba/falcon:speechlunch_june2019_vqvae
- Slides for WaveNet Vocoder: http://www.cs.cmu.edu/~srallaba/ProjectPeregrine/reports/presentation_wavenet.pdf