# TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2020

## Variational Autoencoders (VAEs)

# Latent Variable Models

$$P_\Phi(y) = \sum_z P_\Phi(z) P_\Phi(y|z) = E_{z \sim P_\Phi(z)} P_\Phi(y|z)$$

Or

$$P_\Phi(y|x) = \sum_z P_\Phi(z|x) P_\Phi(y|z,x) = E_{z \sim P_\Phi(z|x)} P_\Phi(y|z,x)$$

Here $z$ is a latent variable.

## Latent Variable Models

$$P_\Phi(y) = \sum_z P_\Phi(z) P_\Phi(y|z) = E_{z \sim P_\Phi(z)} P_\Phi(y|z)$$

Here we often think of $z$ as the causal source of $y$.

For example $z$ might be a physical scene causing image $y$.

Or $z$ might be the intended utterance causing speech signal $y$.

In these situations a latent variable model should more accurately represents the distribution on $y$.

# Latent Variable Models

$$P_\Phi(y) = \sum_z P_\Phi(z) P_\Phi(y|z) = E_{z \sim P_\Phi(z)} P_\Phi(y|z)$$

$P_\Phi(z)$ is called the prior.

Given an observation of $y$ (the evidence) $P_\Phi(z|y)$ is called the posterior.

Variational Bayesian inference involves approximating the posterior.

# Colorization with Latent Segmentation


$x$ $\hat{y}$ $y$

Larsson et al., 2016

Colorization is a natural self-supervised learning problem — we delete the color and then try to recover it from the grey-level image.

Can colorization be used to learn segmentation?

Segmentation is latent — not determined by the color label.

# Colorization with Latent Segmentation


$x$ $\hat{y}$ $y$

Larsson et al., 2016

$x$ is a grey level image.

$y$ is a color image drawn from $\text{Pop}(y|x)$.

$\hat{y}$ is an arbitrary color image.

$P_\Phi(\hat{y}|x)$ is the probability that model $\Phi$ assigns to the color image $\hat{y}$ given grey level image $x$.

# Colorization with Latent Segmentation

Input | Our Method | Ground-truth

$x$ $\hat{y}$ $y$

$$P_\Phi(\hat{y}|x) = \sum_z P_\Phi(z|x) P_\Phi(\hat{y}|z,x).$$

input $x$

$P_\Phi(z|x) = \ldots$ semantic segmentation

$P_\Phi(\hat{y}|z,x) = \ldots$ segment colorization

## Assumptions

We assume models $P_\Phi(z)$ and $P_\Phi(y|z)$ are both samplable and computable.

In other words, we can sample from these distributions and for any given $z$ and $y$ we can compute $P_\Phi(z)$ and $P_\Phi(y|z)$.

These are nontrivial assumptions.

A loopy graphical model is neither (efficiently) samplable nor computable.

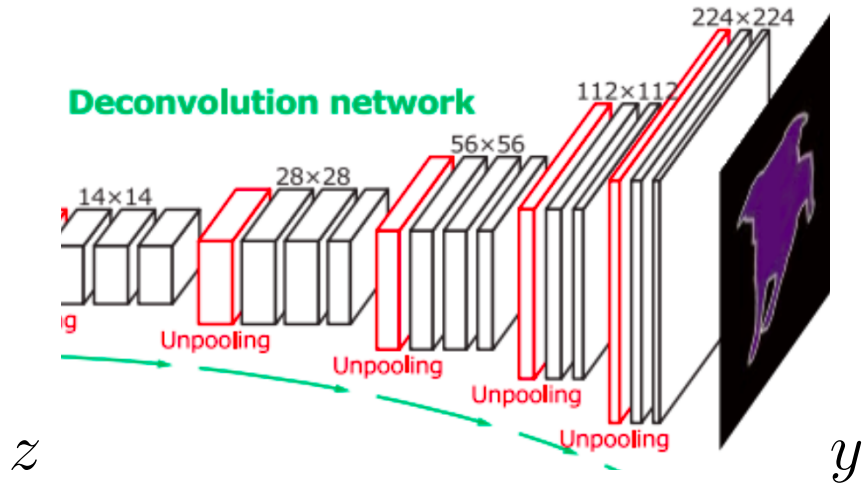
## Cases Where the Assumptions Hold

In CTC we have that $z$ is the sequence with blanks and $y$ is the result of removing the blanks from $z$.

In a hidden markov model $z$ is the sequence of hidden states and $y$ is the sequence of emissions.

An autoregressive model, such as an autoregressive language model, is both samplable and computable.

# Image Generators




We can generate an image $y$ form noise $z$ where $p_\Phi(z)$ and $p_\Phi(y|z)$ are both samplable and computable.

Typically $p_\Phi(z)$ is $\mathcal{N}(0, I)$ reshaped as $z[X, Y, J]$

## Assumptions

Even when $P_\Phi(z)$ and $P_\Phi(y|z)$ are samplable and computable we cannot typically compute $P_\Phi(y)$.

Specifically, for $P_\Phi(y)$ defined by a GAN generator we cannot compute $P_\Phi(y)$ for a test image $y$.

Hence it is not obvious how to optimize the fundamental equation.

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \; E_{y \sim \mathrm{Pop}} \; - \ln P_\Phi(y)$$

# The Evidence Lower Bound (ELBO)

$$
\begin{aligned}
P_\Phi(y) &= \frac{P_\Phi(y)P_\Phi(z|y)}{P_\Phi(z|y)} \\
&= \frac{P_\Phi(z)P_\Phi(y|z)}{P_\Phi(z|y)} \\
\ln P_\Phi(y) &= \ln \frac{P_\Phi(z)P_\Phi(y|z)}{P_\Phi(z|y)} \\
&= E_{z\sim Q_\Phi(z|y)} \ln \frac{P_\Phi(z)P_\Phi(y|z)}{P_\Phi(z|y)}
\end{aligned}
$$

## The Evidence Lower Bound (The ELBO)

We introduce a samplable and computable model $Q_\Phi(z|y)$ to approximate $P_\Phi(z|y)$.

$$
\begin{aligned}
\ln P_\Phi(y) &= E_{z\sim Q_\Phi(z|y)} \ln \frac{P_\Phi(z)P_\Phi(y|z)}{P_\Phi(z|y)} \\
&= E_{z\sim Q_\Phi(z|y)} \left( \ln \frac{P_\Phi(z)P_\Phi(y|z)}{Q_\Phi(z|y)} + \ln \frac{Q_\Phi(z|y)}{P_\Phi(z|y)} \right) \\
&= \left( E_{z\sim Q_\Phi(z|y)} \ln \frac{P_\Phi(z)P_\Phi(y|z)}{Q_\Phi(z|y)} \right) + KL(Q_\Phi(z|y), P_\Phi(z|y)) \\
&\geq E_{z\sim Q_\Phi(z|y)} \ln \frac{P_\Phi(z)P_\Phi(y|z)}{Q_\Phi(z|y)} \quad \text{The ELBO}
\end{aligned}
$$

# The Variational Auto-Encoder (VAE)

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \; E_{y \sim \mathrm{Pop},\, z \sim Q_\Phi(z|y)} \; - \ln \frac{P_\Phi(z) P_\Phi(y|z)}{Q_\Phi(z|y)}$$

# VAE generalizes EM

Expectation Maximimization (EM) applies in the (highly special) case where the exact posterior $P_\Phi(z|y)$ is samplable and computable. EM alternates exact optimization of $Q$ and $P$.

$$\text{VAE: } \Phi^* = \text{argmin}_\Phi \, E_{y\sim\text{Train},\, z\sim Q_\Phi(z|y)} \quad -\ln \frac{P_\Phi(z,y)}{Q_\Phi(z|y)}$$

$$\text{EM: } \Phi^{t+1} = \text{argmin}_\Phi \;\; E_{y\sim\text{Train}} \, E_{z\sim P_{\Phi^t}(z|y)} \;\; -\ln P_\Phi(z,y)$$

| Update (M Step) | Inference (E Step) |
|---|---|
| Hold $Q$ fixed | $Q(z\|y) = P_{\Phi^t}(z\|y)$ |

## Hard EM relies on Closed Form $Q^*$

EM: $\Phi^{t+1} = \text{argmin}_\Phi \; E_{y,z\sim P_{\Phi^t}(z|y)} \; -\ln P_\Phi(z,y)$

Hard EM: $\Phi^{t+1} = \text{argmin}_\Phi \; E_{y,z=\text{argmax}_z P_{\Phi^t}(z|y)} \; -\ln P_\Phi(z,y)$

This relies on $P_{\Phi^t}(z|y)$ being exactly computable so that the optimization over $Q$ in VAE has a closed form solution.

For a "hard VAE" we need some way of training $Q$ other than sampling from it (see the slides on VQ-VAE).

## The Reparameterization Trick

$$-\ln P_\Phi(y) \leq E_{z\sim Q_\Phi(z|y)} \; -\ln \frac{P_\Phi(z)P_\Phi(y|z)}{Q_\Phi(z|y)}$$

$$= E_\epsilon \; -\ln \frac{P_\Phi(z)P_\Phi(y|z)}{Q_\Phi(z|y)} \quad z := f_\Phi(y,\epsilon)$$

$\epsilon$ is parameter-independent noise.

This supports SGD: $\nabla_\Phi \, E_{y,\epsilon} \, [\ldots] = E_{y,\epsilon} \, \nabla_\Phi \, [\ldots]$

# Gaussian VAEs

$$\Phi^* = \operatorname*{argmin}_{\Phi} E_{y,\epsilon} \; - \ln \frac{p_\Phi(z) p_\Phi(y|z)}{q_\Phi(z|y)}$$

$$z = z_\Phi(y) + \sigma_\Phi(y) \odot \epsilon \quad \epsilon \sim \mathcal{N}(0, I)$$

$$q_\Phi(z[i]|y) = \mathcal{N}(z_\Phi(y)[i], \sigma_\Phi(y)[i])$$

$$p_\Phi(z[i]) = \mathcal{N}(\mu_p, \sigma_p[i]) \;\; \text{WLOG} = \mathcal{N}(0, 1)$$

$$p_\Phi(y|z) = \mathcal{N}(y_\Phi(z), \; \sigma^2 I)$$

## $-\ln p_\Phi(y|z)$ as Distortion

For $p_\Phi(y|z) \propto \exp(-||y - y_\Phi(z)||^2/(2\sigma^2))$ we get

$$\begin{aligned}\Phi^* &= \underset{\Phi}{\operatorname{argmin}}\, E_{y,\epsilon} \quad -\ln \frac{p_\Phi(z)}{q_\Phi(z|y)} - \ln p_\Phi(y|z) \\ &= \underset{\Phi,\sigma}{\operatorname{argmin}}\, E_{y,\epsilon} \quad -\ln \frac{p_\Phi(z)}{q_\Phi(z|y)} + \left(\frac{1}{2\sigma^2}\right) ||y - y_\Phi(z)||^2 + d \ln \sigma\end{aligned}$$

where

$d$ is the dimension of $y$ and $\sigma^* = \sqrt{\frac{1}{d}\, E_{y,\epsilon}\, ||y - y_\Phi(z)||^2}$

## Posterior Collapse

Assume Universal Expressiveness for $P_\Phi(y|z)$.

This allows $P_\Phi(y|z) = \text{Pop}(y)$ independent of $z$.

We then get a completely optimized model with $z$ taking a single (meaningless) determined value.

$$Q_\Phi(z|y) = P_\Phi(z|y) = 1$$

## Colorization with Latent Segmentation


$x$ $\hat{y}$ $y$

Larsson et al., 2016

Can colorization be used to learn latent segmentation?

We introduce a latent segmentation into the model.

In practice the latent segmentation is likely to "collapse" because the colorization can be done just as well without it.

## Independent Universality

$$\Phi^* = \operatorname*{argmin}_{\Phi} E_{y \sim \text{Pop},\, z \sim Q_\Phi(z|y)} \quad - \ln \frac{P_\Phi(z, y)}{Q_\Phi(z|y)}$$

It is natural to assume that $\Phi$ has independent parameters for each distribution. In practice parameters are often shared.

Since $\Phi$ can indepedently parameterize each distribution, we will often use an independent universality assumption that $\Phi$ can represent any triple of distributions $Q(z|y)$, $P(z)$ and $P(y|z)$.

## Independent Universality

More formally, we will often assume that for any triple of distributions $Q(z|y)$, $P(z)$ and $P(y|z)$ there exists a $\Phi$ that simultaneously satisfies

$$\begin{aligned} Q_\Phi(z|y) &= Q(z|y) \\ P_\Phi(z) &= P(z) \\ P_\Phi(y|z) &= P(y|z) \end{aligned}$$

This assumption allows each distribution to be independently optimized while holding the others fixed.

# The $\beta$-VAE

$\beta$-VAE: Learning Basic Visual Concepts With A Constrained Variational Framework, Higgins et al., ICLR 2017.

The $\beta$-VAE introduces a parameter $\beta$ allow control of the rate-distortion trade off.

# Indeterminacy of the VAE

$$\text{VAE: } \Phi^* = \text{argmin}_\Phi \, E_{y,\epsilon} \; - \ln \frac{P_\Phi(z)}{Q_\Phi(z|y)} - \ln P_\Phi(y|z)$$

Assuming independent universality we can optimize $P_\Phi(z)$ and $P_\Phi(y|z)$ while holding $Q_\Phi(z|y)$ fixed. This gives

$$P^*(z) \; = \; = P_{\text{pop}}(z) = E_y \, Q_\Phi(z|y)$$

$$P^*(y|z) \; = \; P^*_{\text{pop}}(y|z) \propto P(y,z) = \text{Pop}(y) Q_\Phi(z|y)$$

## Indeterminacy of the VAE

$$\begin{aligned}
\Phi^* &= \operatorname*{argmin}_{\Phi} E_{y,\epsilon} \; - \ln \frac{P_{\text{pop}}(z)}{Q_\Phi(z|y)} - \ln P_{\text{pop}}(y|z) \\
&= \operatorname*{argmin}_{\Phi} \quad I_\Phi(y,z) \; + \; H_\Phi(y|z) \\
&= \operatorname*{argmin}_{\Phi} \qquad H_{\text{pop}}(y)
\end{aligned}$$

But $H_{\text{pop}}(y)$ is independent of $\Phi$.

Any choice of $Q_\Phi(z|y)$ gives optimal modeling of $y$.

## Indeterminacy of the VAE

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \; I_\Phi(y, z) + H_\Phi(y|z)$$

The choice of $Q_\Phi(z|y)$ does not influence the value of the objective function but controls $I(y, z)$.

We have $0 \leq I(y, z) \leq H(y)$ with the full range possible.

## The $\beta$-VAE

To control $I(y, z)$ we introduce a weighting $\beta$

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \ \beta I_\Phi(y, z) + H_\Phi(y|z)$$

$$\beta\text{-VAE} \ \ \Phi^* = \underset{\Phi}{\operatorname{argmin}} \, E_{y,\epsilon} \ - \beta \ln \frac{P_\Phi(z)}{Q_\Phi(z|y)} - \ln P_\Phi(y|z)$$

For $\beta < 1$ we no longer have an upper bound on $H_{\text{pop}}(y)$ but we can force the use of $z$ (avoid posterior collapse).

For $\beta > 1$ the bound on $H_{\text{Pop}}(y)$ becomes weaker and the latent variables carry less information.

# RDAs vs. $\beta$-VAEs

RDAs and $\beta$-VAEs are essentially the same.

$$\text{RDA: } \Phi^* = \text{argmin}_\Phi \, E_{y,z\sim Q_\Phi(z|y)} \, -\ln \frac{P_\Phi(z)}{Q_\Phi(z|y)} + \lambda \text{Dist}(y, y_\Phi(z))$$

$$\beta\text{-VAE: } \Phi^* = \text{argmin}_\Phi \, E_{y,z\sim Q_\Phi(z|y)} \, -\beta \ln \frac{P_\Phi(z)}{Q_\Phi(z|y)} - \ln P_\Phi(y|z)$$

# VAEs 2013

Sample $z \sim \mathcal{N}(0, I)$ and compute $y_\Phi(z)$

[Alec Radford]

# VAEs 2019

VQ-VAE-2, Razavi et al. June, 2019

# VAEs 2019

Figure 1: Class-conditional 256x256 image samples from a two-level model trained on ImageNet.

VQ-VAE-2, Razavi et al. June, 2019

## Vector Quantized VAEs (VQ-VAE)

VQ-VAEs effectively perform $k$-means on vectors in the model so as to represent vectors by discrete cluster centers.

For concreteness we will consider VQ-VAEs on images with a single layer of quantization.

We use $x$ and $y$ for spatial image coordinates and use $s$ (for signal) to denote images.

## VQ-VAE Encoder-Decoder

We train a dictionary $C[K, I]$ where $C[k, I]$ is the center vector of cluster $k$.

$$L[X, Y, I] = \text{Enc}_\Phi(s)$$

$$z[x, y] = \underset{k}{\text{argmin}} \ ||L[x, y, I] - C[k, I]||$$

$$\hat{L}[x, y, I] = C[z[x, y], I]$$

$$\hat{s} = \text{Dec}_\Phi(\hat{L}[X, Y, I])$$

The "symbolic image" $z[X, Y]$ is the latent variable.

# VQ-VAE as an RDA

We will interpret the VQ-VAE as an RDA.

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}}\, E_s\, I(s,z) + \lambda \mathrm{Dist}(s,\hat{s})$$

The mutual information $I(s,z)$ is limited by the entropy of $z[X,Y]$ which can be no larger than $\ln K^{XY} = XY \ln K$.

Maximizing $I(s,z)$ subject to this upper bound should reduce the distortion by providing the decoder with adequate information about the image.

# VQ-VAE Training Loss

We preserve information about the image $s$ by minimizing the distortion between $L[X, Y, I]$ and its reconstruction $\hat{L}[X, Y, I]$.

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \; E_s \; \beta||L[X, Y, I] - \hat{L}[X, Y, I]||^2 + ||s - \hat{s}||^2$$

This is a two-level rate-distortion auto-encoder where the rate can be no larger than $XY \ln K$.

## Parameter-Specific Learning Rates

$$||L[X,Y,I] - \hat{L}[X,Y,I]||^2 = \sum_{x,y} ||L[x,y,I] - C[z[x,y],I]||^2$$

For the gradient of this they use

$$\begin{aligned} \text{for } x,y \;\; L[x,y,I].\text{grad} \;\texttt{+=}\; 2\beta(L[x,y,I] - C[z[x,y],I]) \\ \text{for } x,y \;\; C[z[x,y],I].\text{grad} \;\texttt{+=}\; 2(C[z[x,y],I] - L[x,y,I]) \end{aligned}$$

This gives a parameter-specific learning rate for $C[K,I]$.

Parameter-specific learning rates do not change the stationary points (the points where the gradients are zero).

## The Relationship to $K$-means

for $x, y$ $C[z[x, y], I]$.grad += $2(C[z[x, y], I] - L[x, y, I])$

At a stationary point we get that $C[k, I]$ is the mean of the set of vectors $L[x, y, I]$ with $z[x, y] = k$ (as in $K$-means).

## Straight Through Gradients

The latent variables are discrete so some approximation to SGD must be used.

They use "straight-through" gradients.

$$\text{for } x, y \;\; L[x, y, I].\text{grad } \texttt{+=} \; \hat{L}[x, y, I].\text{grad}$$

This assumes low distortion between $L[X, Y, I]$ and $\hat{L}[X, Y, I]$.

## A Suggested Modification

The parameter $\beta$ is paying two roles

- It controls the relative weight of the two distortion losses.
- It controls the learning rate adjustment for the codebook.

Shouldn't we have separate parameters for these two roles?

## Training Phase II

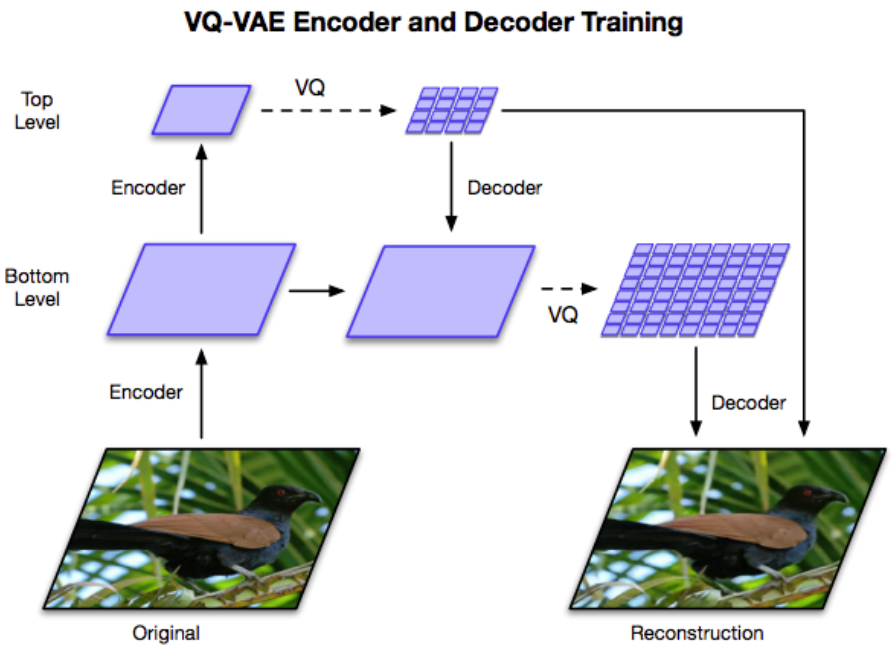
Once the model is trained we can sample images $s$ and compute the "symbolic image" $z[X, Y]$.

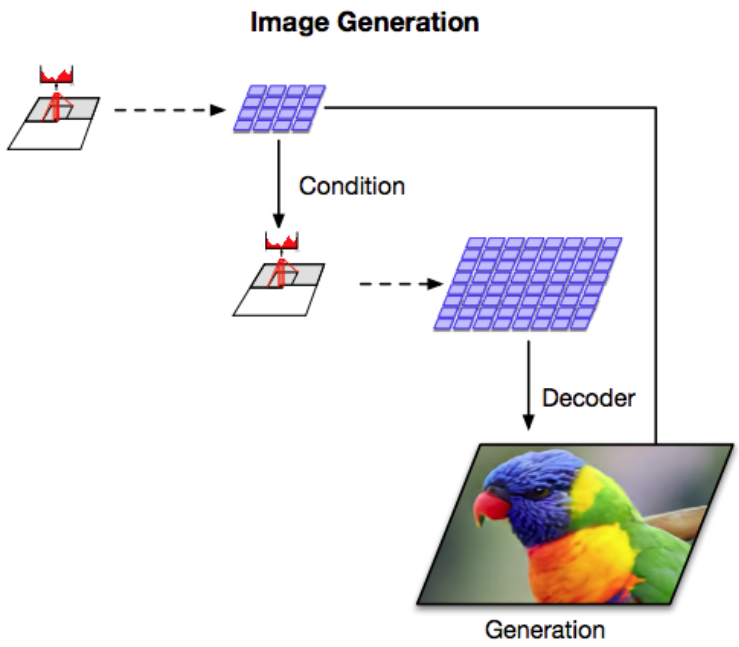
Given samples of symbolic images $z[X, Y]$ we can learn an auto-regressive model of these symbolic images using a pixel-CNN.

This yields a prior probability distribution $P_\Phi(z[X, Y])$ which provides a tighter upper bound on the rate.

We can then measure compression and distortion for test images. This is something GANs cannot do.

# Multi-Layer Vector Quantized VAEs



## Quantitative Evaluation

The VQ-VAE2 paper reports a classification accuracy score (CAS) for class-conditional image generation.

We generate image-class pairs from the generative model trained on the ImageNet training data.

We then train an image classifier from the generated pairs and measure its accuracy on the ImageNet test set.

| | Top-1 Accuracy | Top-5 Accuracy |
|---|---|---|
| BigGAN deep | 42.65 | 65.92 |
| VQ-VAE | 54.83 | 77.59 |
| VQ-VAE after reconstructing | 58.74 | 80.98 |
| Real data | 73.09 | 91.47 |

## Direct Rate-Distortion Evaluation.

Rate-distortion metrics for image compression to discrete representations support unambiguous rate-distortion evaluation.

Rate-distortion metrics also allow one to explore the rate-distortion trade-off.

| | Train NLL | Validation NLL | Train MSE | Validation MSE |
|---|---|---|---|---|
| Top prior | 3.40 | 3.41 | - | - |
| Bottom prior | 3.45 | 3.45 | - | - |
| VQ Decoder | - | - | 0.0047 | 0.0050 |

Table 1: Train and validation negative log-likelihood (NLL) for top and bottom prior measured by encoding train and validation set resp., as well as Mean Squared Error for train and validation set. The small difference in both NLL and MSE suggests that neither the prior network nor the VQ-VAE overfit.

# Image Compression

$h_{\text{top}}$ $h_{\text{top}}, h_{\text{middle}}$ $h_{\text{top}}, h_{\text{middle}}, h_{\text{bottom}}$ Original

Figure 3: Reconstructions from a hierarchical VQ-VAE with three latent maps (top, middle, bottom). The rightmost image is the original. Each latent map adds extra detail to the reconstruction. These latent maps are approximately 3072x, 768x, 192x times smaller than the original image (respectively).

## Vector Quantization (Emergent Symbols)

Vector quantization represents a distribution (or density) on vectors with a discrete set of embedded symbols.

Vector quantization optimizes a rate-distortion tradeoff for vector compression.

The VQ-VAE uses vector quantization to construct a discrete representation of images and hence a measurable image compression rate-distortion trade-off.

## Symbols: A Better Learning Bias

Do the objects of reality fall into categories?

If so, shouldn't a learning architecture be designed to categorize?

Whole image symbols would yield emergent whole image classification.

## Symbols: Improved Interpretability

Vector quantization shifts interpretation from linear threshold units to the emergent symbols.

This seems related to the use of t-SNE as a tool in interpretation.

## Symbols: Unifying Vision and Language

Modern language models use word vectors.

Word vectors are embedded symbols.

Vector quantization also results in models based on embedded symbols.

## Symbols: Addressing the "Forgetting" Problem

When we learn to ski we do not forget how to ride a bicycle.

However, when a model is trained on a first task, retraining on a second tasks degrades performance on the first (the model "forgets").

But embedded symbols can be task specific.

The embedding of a task-specific symbol will not change when training on a different task.

## Symbols: Improved Transfer Learning.

Embedded symbols can be domain specific.

Separating domain-general parameters from domain-specific parameters may improve transfer between domains.

## Unsupervised Machine Translation

We can treat the German sentence $z$ as a latent variable in a probability model of a English sentence $y$.

$$\Phi^* = \operatorname*{argmin}_{\Phi} \; E_{y,z\sim Q_\Phi(z|y)} \quad -\beta \ln \frac{P_\Phi(z)}{Q_\Phi(z|y)} - \ln P_\Phi(y|z)$$

Here $P_\Phi(z)$ can be a trained language model for German and $P_\Phi(y|z)$ and $Q_\Phi(z|y)$ are translation models.

# Unsupervised Machine Translation

In practice we use "backtranslation"

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \begin{array}{ll} & E_{y\sim \mathrm{Pop}_y,\, z\sim Q_\Phi(z|y)} \quad -\ln P_\Phi(y|z) \\ + & E_{z\sim \mathrm{Pop}_z,\, y\sim P_\Phi(y|z)} \quad -\ln Q_\Phi(z|y) \end{array}$$

# END