University of Illinois Fall 2018

ECE 544NA: Pattern Recognition
Lecture 20: November 1

Lecturer: Alexander Schwing Scribe: Siqi Zhang

1 Overview

In just three years, Variational Auto-Encoders (VAEs) have emerged as one of the most popular approaches to unsupervised learning of complicated distributions. VAEs are appealing because they are built on top of standard function approximators (neural networks), and can be trained with stochastic gradient descent [1].

Keywords: variational autoencoders, unsupervised learning, structured prediction, neural networks.

1.1 Goals of this lecture

- Getting to know Variational Auto-Encoders.
- Understanding generative methods.
- Differentiating between discriminative and generative methods.

1.2 Recap

Maximum likelihood so far?
Model:

$$p(y \mid x) = \frac{\exp F(y, x, w)/\varepsilon}{\sum_{\hat{y}} \exp F(\hat{y}, x, w)/\varepsilon} \tag{1}$$

- **y**: discrete output space
- x: input data

How about modeling a distribution **p(x)** for the data given data points?

Generative Models: Given training data, generate new samples from same distribution.

- Training data $\sim$ p$_{model}$(x).
- Generated samples $\sim$ p$_{data}$(x).

Want to learn p$_{model}$(x) similar to p$_{data}$(x). How to model p(x)?

- Fit mean and variance, parameters θ, of a distribution (e.g.,Gaussian).
- Fit parameters θ of a mixture distribution (e.g., mixture of Gaussian, k-means).

2 Variational Auto-Encoders

2.1 Visualisation: MNIST handwritten digits dataset

Input: An image in $\mathbb{R}^{28\times 28}$.
Latent space: An input in $\mathbb{R}^{28\times 28}$ doesn't explicitly contain the single digit (0-9) information. It resides in the latent space.
You can think of the latent space as $\mathbb{R}^k$ where every vector contains k pieces of essential information needed to draw an image [2]. For example, the first dimension contains the number represented by the digit, the second being the width, and the third being the angle.
Intuitively, it helps if the model first decides which character to generate before it assigns a value to any specific pixel. This kind of decision is formally called a *latent variable*.
How VAE models the process of generating images? We will cover this in section 2.8.

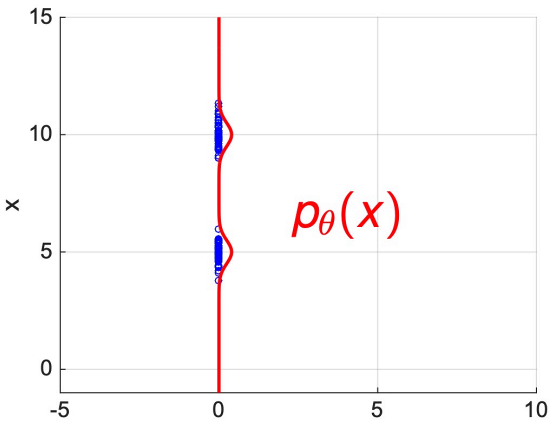

Figure 1: Fit parameters θ of a mixture distribution

How can we learn $p_\theta(x)$?

$$p_\theta(x) = \int p_\theta(z) p_\theta(x \mid z) dz$$

where $p_\theta(z)$ is simple gaussian prior. Imposing a gaussian distribution serves for training purposes only, then we will be able to train the model using gradient descent.
Now we look into $p_\theta(x \mid z)$.

- x is a high dimensional vector.
- data is concentrated around a low dimensional manifold.

Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data.
Q: Why dimensionality reduction?
A: Want features to capture meaningful factors of variation in data [3].
Autoencoders can reconstruct data, and can learn features to initialize a supervised model.

We can learn $p_\theta(x \mid z)$ by decoder neural network.

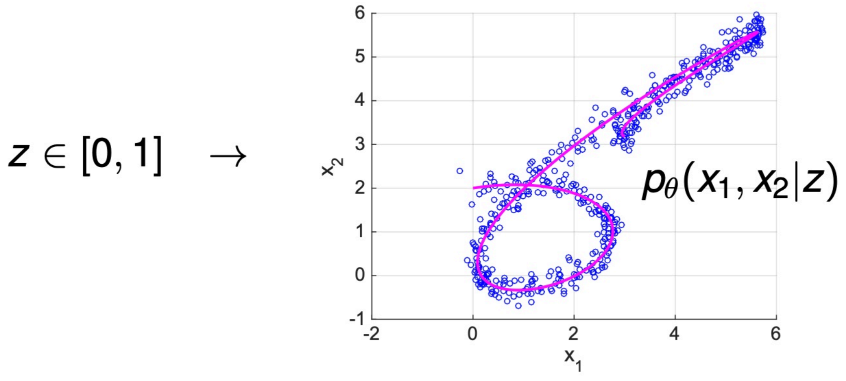

Figure 2: Low-dimensional manifolds (z $\in \mathbb{R}^2$)

2.2 Background: auto-encoder

An autoencoder network is actually a pair of two connected networks, an encoder and a decoder. An encoder network takes in an input, and converts it into a smaller, dense representation, which the decoder network can use to convert it back to the original input.

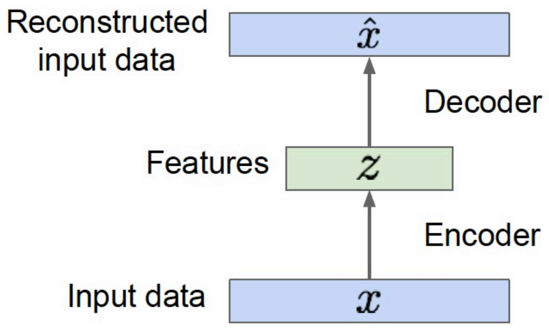

Figure 3: Training as an auto-encoder

2.3 Variational auto-encoder architecture: Decoder

VAEs define intractable density function with latent **z**. We aim at maximizing the probability of each x in the training set under the entire generative process, in other words, estimating the true parameters θ of this generative model to maximize likelihood of training data, according to:

$$p_\theta(x) = \int p_\theta(z) p_\theta(x \mid z) dz \tag{2}$$

Here $p_\theta(x \mid z)$ allows us to make the dependence of x on z explicit by using the law of total probability. What is $p_\theta(x \mid z)$ then?
The intuition behind this framework—called "maximum likelihood"— is that if the model is likely to produce training set samples, then it is also likely to produce similar samples, and unlikely to produce dissimilar ones. The important property is simply that $p_\theta(x \mid z)$ can be computed, and is

continuous in θ.
To solve Equation 2, there are two problems that VAEs must deal with: how to define the latent variables z (i.e., decide what information they represent), and how to deal with the integral over z.

Idea: z from simple Gaussian & transformation via deep net.
Provided powerful function approximators, we can simply learn a function which maps our independent, normally-distributed z values to whatever latent variables might be needed for the model, and then map those latent variables to x.

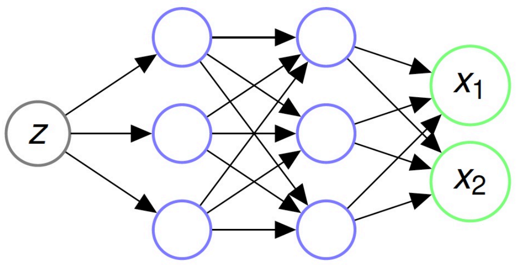

Figure 4: Decoder($p_\theta(x \mid z)$)

Choosing prior p(z) to be simple, e.g. Gaussian, is reasonable for latent attributes. Transformation via deep net.
But how to optimize for θ? How do we know which z maps to which x? We need to learn $p_\theta(z \mid x)$ to figure out.

2.4 Variational auto-encoder architecture: Encoder

$$p_\theta(z \mid x) = \frac{p_\theta(x \mid z)p(z)}{p_\theta(x)} = \frac{p_\theta(x \mid z)p(z)}{\int_{\hat{z}} p_\theta(x \mid \hat{z})p(\hat{z})d\hat{z}} \tag{3}$$

Notice that the integral in the denominator is **intractable**, thus bringing intractable data likelihood.
There are several ways to compute normalization constant, for example, sampling techniques (generally very costly), so we choose to **approximate** $p_\theta(z \mid x)$ **with** $q_\phi(z \mid x)$**) (encoder)** here.
Will see that this allows us to derive a lower bound on the data likelihood that is tractable, which we can optimize.

$$q_\phi(z \mid x) = \mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x)) \tag{4}$$

Learning parameters ϕ, θ via back-propagation. Then we can find the best ϕ, θ by optimizing the loss function.
Note: In order to deal with the fact that the network may learn negative values for σ, we have the network learn $\log \sigma^2$.

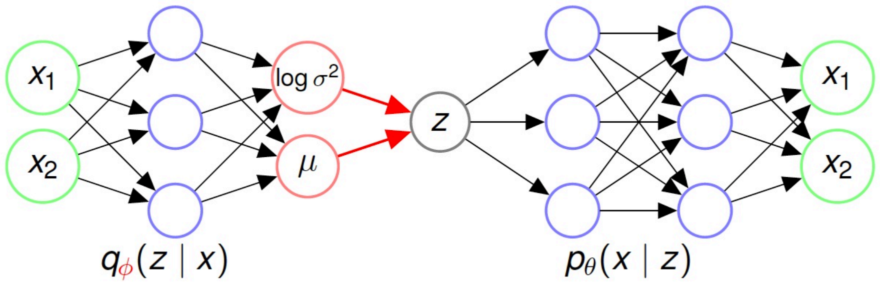

Figure 5: VAE architecture

2.5 The loss function

The (log) data likelihood:

$$\begin{aligned}
\log p_\theta(x) &= \int_z q_\phi(z \mid x) \log p_\theta(x) \\
&= \int_z q_\phi(z \mid x) \log \frac{p_\theta(x,z)}{p_\theta(z \mid x)} \\
&= \int_z q_\phi(z \mid x) \log \left(\frac{p_\theta(x,z)}{q_\phi(z \mid x)} \cdot \frac{q_\phi(z \mid x)}{p_\theta(z \mid x)} \right) \\
&= \int_z q_\phi(z \mid x) \log \frac{p_\theta(x,z)}{q_\phi(z \mid x)} + \int_z q_\phi(z \mid x) \log \frac{q_\phi(z \mid x)}{p_\theta(z \mid x)} \\
&= L(p_\theta, q_\phi) + D_{KL}(q_\phi, p_\theta) \\
&\geq L(p_\theta, q_\phi)
\end{aligned} \tag{5}$$

L there is often referred to as empirical lower bound (ELBO).
We can't compute this KL term, but we know KL divergence always ≥ 0, there comes our lower bound L, and it will be the loss function then. The loss function of VAE is usually either the mean-squared error or cross-entropy between the output and the input, known as the *reconstruction loss*, which penalizes the network for creating outputs different from the input.

2.6 Approximate Inference

The tractable lower bound can be further written as:

$$\begin{aligned}
L(p_\theta, q_\phi) &= \int_z q_\phi(z \mid x) \log \frac{p_\theta(x,z)}{q_\phi(z \mid x)} \\
&= \int_z q_\phi(z \mid x) \log \frac{p_\theta(x \mid z)p(z)}{q_\phi(z \mid x)} \\
&= \int_z q_\phi(z \mid x) \log \frac{p(z)}{q_\phi(z \mid x)} + \int_z q_\phi(z \mid x) \log p_\theta(x \mid z) \\
&= -D_{KL}(q_\phi, p) + \mathbb{E}_{q\phi}[\log p_\theta(x \mid z)]
\end{aligned} \tag{6}$$

where $\mathbb{E}_{q\phi}[\log p_\theta(x \mid z)]$ comes from taking expectation w.r.t. z (using encoder network) and represents the reconstruction likelihood.

KL is the Kullback–Leibler divergence, which intuitively measures how similar two distributions are. $D_{KL}(q_\phi, p)$ measures how off the approximated posterior distribution is from prior. Minimizing the KL divergence here means optimizing the probability distribution parameters ($\mu_\phi(x)$ and $\sigma_\phi(x)$) to closely resemble that of the target distribution. The KL divergence is minimized when $\mu_\phi(x) = 0$ and $\sigma_\phi(x) = 1$.

1. In one hand we want to maximize how well x is expected to be decoded from $z \sim q_\phi(z \mid x)$.
2. On the other hand, we want q_ϕ (the encoder) to be similar to the prior p(z) (a multivariate Gaussian). One can think of this term as a regularization.

- Reconstruction
  Approximate $\mathbb{E}_{q\phi}[\log p_\theta(x \mid z)]$ by sampling from $q_\phi(z \mid x)$.

$$\begin{aligned}\mathbb{E}_{q\phi}[\log p_\theta(x \mid z)] &= \int_z q_\phi(z \mid x) \log p_\theta(x \mid z) \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(x \mid z^i)\end{aligned} \tag{7}$$

There we come to the variational auto-encoder **loss function**:

$$L(p_\theta, q_\phi) \approx -D_{KL}(q_\phi, p_\theta) + \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(x \mid z^i) \tag{8}$$

where $z \sim q_\phi(z \mid x) = \mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x))$.

2.7 Reparameterization trick

How to backpropagate through the sampling step? Notitec that

$$z \sim q_\phi(z \mid x) = \mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x))$$

is equivalent to

$$z = \mu_\phi(x) + \sigma_\phi(x) \cdot \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, 1)$$

We randomly sample ϵ from a unit Gaussian, and then shift the randomly sampled ϵ by the latent distribution's mean $\mu_\phi(x)$ and scale it by the latent distribution's variance $\sigma_\phi(x)$. With this reparameterization, we can now optimize the parameters of the distribution while still maintaining the ability to randomly sample from that distribution.
The result will have a distribution equal to $q_\phi(z \mid x)$. Now the sampling operation will be from the standard Gaussian. Hence, the gradients will be able to propagate through $\mu_\phi(x)$ and $\sigma_\phi(x)$, since these are deterministic paths now [4].

2.8 Flowchart

Summarizing all the steps one needs to grasp in order to implement VAE.

1. An input image is passed through an encoder network.
2. The encoder outputs parameters of a distribution $q_\phi(z \mid x)$.

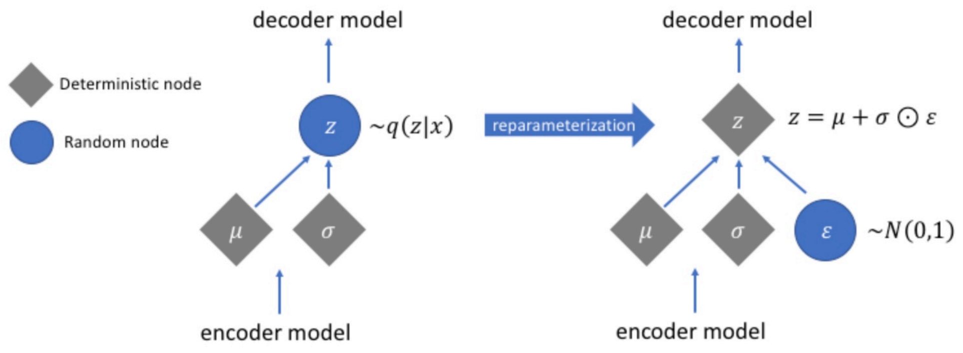

Figure 6: Reparameterization trick

3. A latent vector z is sampled from $q_\phi(z \mid x)$. If the encoder learned to do its job well, most chances are z will contain the information describing x.

4. The decoder decodes z into an image.

5. Reconstruction error: the output should be similar to the input.

6. $q_\phi(z \mid x)$ should be similar to the prior (multivariate standard Gaussian).

Usually, we will constrain the network to produce latent vectors having entries that follow the unit normal distribution. Then, when trying to generate data, we can simply sample some values from this distribution, feed them to the decoder, and the decoder will return us completely new objects that appear just like the objects our network has been trained with.

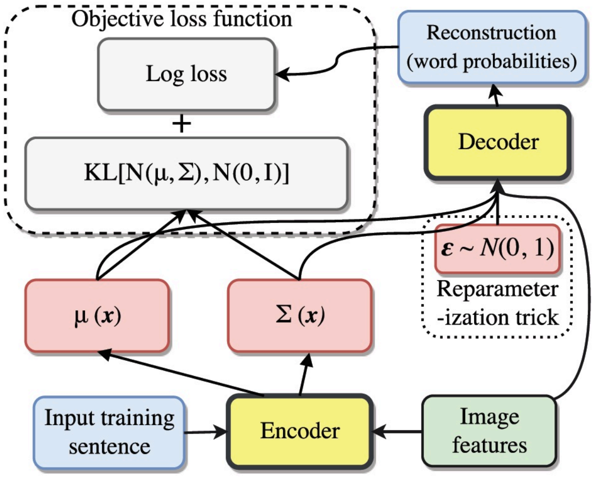

Figure 7: VAE Flowchart

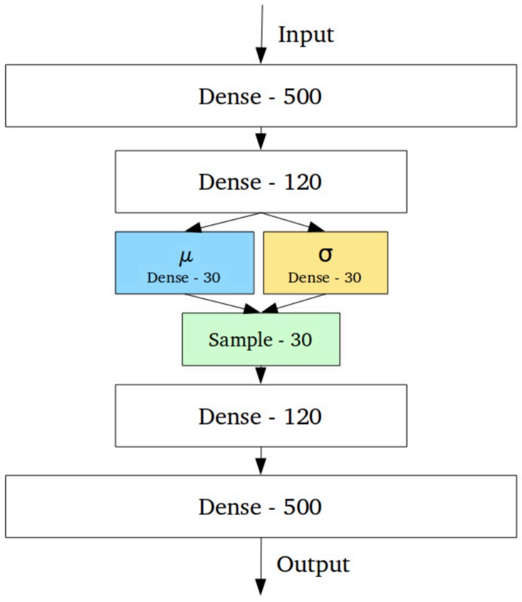

Figure 8: Variational Auto-Encoder

3 Warp Up

3.1 Quiz

- What is the difference between generative and discriminative modeling?
  1. Generative: Assume some functional form for P(X|Y), P(Y). Estimate parameters of P(X|Y), P(Y) directly from training data. Use Bayes rule to calculate P(Y|X=x).
  2. Discriminative: Directly assume some functional form for P(Y|X). Estimate parameters of P(Y|X) directly from training data.
- What generative modeling techniques do you know about?
  Variational Auto-Encoder, Generative Adversarial Nets, Boltzmann Machine.
- What are the approximations used in variational auto-encoders?
  Approximate $p_\theta(z \mid x)$ with $q_\phi(z \mid x)$.
- Why are variational auto-encoder results smooth?
  We're enforcing a continuous, smooth latent space representation. Thus, values which are nearby to one another in latent space should correspond with very similar reconstructions.

3.2 Topics covered

- Getting to know variational auto-encoders, a generative modeling technique.
- Understanding the reasons for approximations.

3.3 Advantage and disadvantage of VAE

Advantages:

- Principled approach to generative models.
- Allows inference of $q_\phi(z \mid x)$, can be useful feature representation for other tasks.

Disadvantags:

- Samples blurrier and lower quality compared to state-of-the-art Generative Adversarial Networks (GANs).

Active areas of research:

- More flexible approximations, e.g. richer approximate posterior instead of diagonal Gaussian.
- Incorporating structure in latent variables.

References

[1] C. Doersch, "Tutorial on variational autoencoders." https://arxiv.org/abs/1606.05908, 2016.

[2] A. Datum, "Variational autoencoders explained." http://anotherdatum.com/vae.html, 2018.

[3] F.-F. Li, J. Johnson, and S. Yeung, "Generative models." http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture12.pdf, 2018.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes." https://arxiv.org/abs/1312.6114, 2014.