Variational Autoencoders

Advanced Machine Learning and Artificial Intelligence

Yuri Balasanov

University of Chicago, MScA

Outline of the Session

- Common vs. Variational Autoencoders
- Low-dimensional representation in generative models
- Finding latent space distribution
- Kullback-Leibler Divergence for Gaussian distributions
- Generating new samples with VAE

Sources:

Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Technologies to Build Intelligent Systems, Aurelien Geron, 2017
D. P. Kingma and M. Welling, "Auto-encoding variational bayes," ArXiv Prepr. ArXiv13126114, 2013
https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf
https://news.sophos.com/en-us/2018/06/15/using-variational-autoencoders-to-learn-variations-in-data/

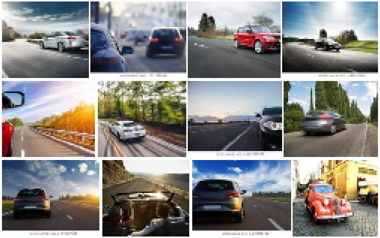
Low-Dimensional Representation and Generative Models I

- Goal of autoencoders is to find a low-dimensional representation of the data
- Look at the picture. It may have very large size
- Low-dimensional representation may be: "Alfa Romeo on the road" (less than 100 bytes)
- Decoding is trained to reconstruct the input as accurate as possible

Figure: Source: https://www.pexels.com/photo/action-asphalt-auto-automobile-210019/

Low-Dimensional Representation and Generative Models II

- Decoding may require knowledge of additional reasonable constraints:
  - General idea of the road
  - Car has 4 wheels and windshield
  - Wheels contact the road, etc.
- Decoding with variety of constraints may result in large variety of images corresponding to "A car on the road"



- Goal of generative models: learn about constraints to generate reasonable new objects: e.g. wheels are in contact with the road

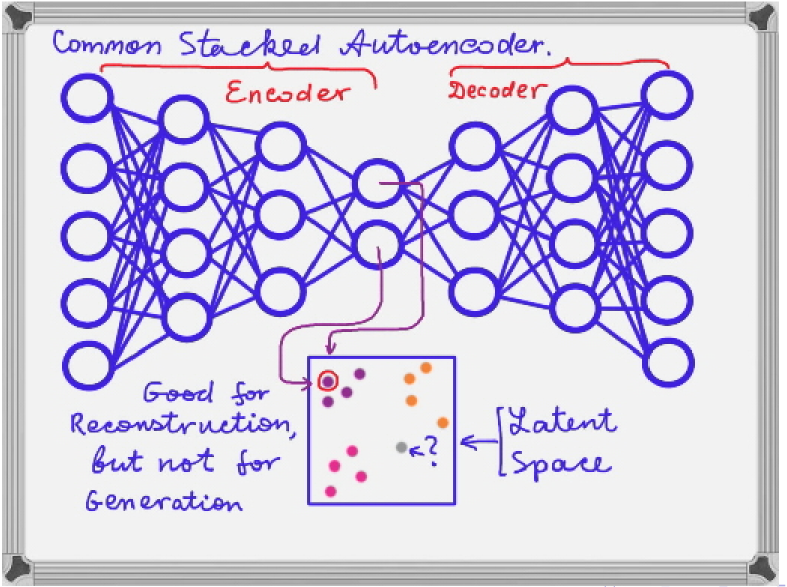
Variational Autoencoders and Their Applications

- **Variational Autoencoders** (**VAE**) belong to a type of **generative models**
- This means that they learn how to code input into a latent probability distribution, generate a sample from it, and then decode the sample vector from the latent distribution into the output similar to input
- Once the model is trained, it can generate new data by sampling from the latent probability distribution and decoding the sample into a new object that the original sample does not contain
- Applications of Variational Autoencoders go from generation of music or human faces not represented in the sample to a general examples of representation learning and semisupervised learning

Comparison of Variational and Common Autoencoders I

- Any autoencoder contains 2 parts:
  - Encoder (**recognition network**), converts inputs to an internal latent representation (coding)
  - Decoder (**generative network**), converts latent coding into the outputs
- Any autoencoder trains both components simultaneously minimizing reconstruction loss
- Any autoencoder is **undercomplete** in some way, typically this means that the number of output units of the encoder is smaller than the number of its inputs

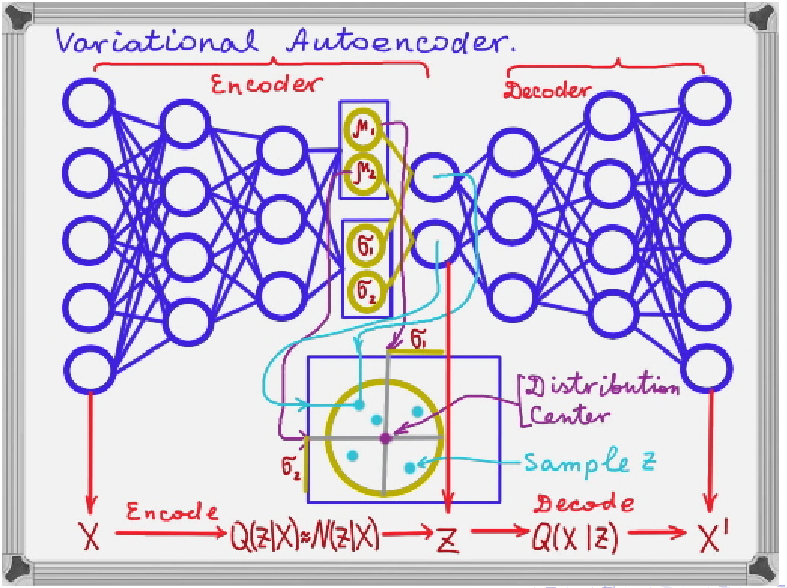
Latent Space of Stacked Autoencoder



Comparison of Variational and Common Autoencoders II

- Codings in the latent space of stacked autoencoder typically have complex multimodal distribution sampling from which is a challenging problem
- Variational autoencoder pulls observations in the latent space to form a latent space distribution called **posterior distribution** which is close to the predefined parametric distribution called **prior distribution**
- Typical prior distribution is a multidimensional isotropic (uncorrelated) Gaussian distribution
- Final stage of encoding in variational autoencoder is:
  - Creation of 2 vectors: $< \mu_1, \dots, \mu_k >$ and $< \sigma_1, \dots, \sigma_k >$, where k is the number of units in bottleneck layer. These are the means and standard deviations of k-dimensional isotropic Gaussian distribution
  - Draw a sample from the isotropic latent space distribution to make codings of the given input
- Note that $< \mu_1, \dots, \mu_k >$ and $< \sigma_1, \dots, \sigma_k >$ are generated for each input in the batch

Latent Space of Variational Autoencoder



Finding Latent Space Distribution I

- Since latent space distribution is a constrained distribution of common stacked autoencoder there should be a modification of the loss function used for training
- The model assumes that the data are generated in two-step process:
  1. Generate Z from a prior distribution $P(Z)$
  2. Generate X from conditional distribution $Q(X|Z)$
- Then posterior distribution $Q(Z|X)$ of the encoder is obtained by Bayes Theorem as

$$Q(Z|X) = \frac{Q(X|Z)\,P(Z)}{P(X)}$$

- Unfortunately, standard Bayesian method of finding posterior $Q(Z|X)$, that is MCMC, does not help because the denominator $P(X)$ is not computationally tractable

Finding Latent Space Distribution II

- Instead of finding posterior $Q(Z|X)$ with MCMC introduce a recognition model $N(Z|X)$ also called **probabilistic encoder**, a parametric approximation to the posterior
- Probabilistic encoder for each input X produces a distribution of the latent variable Z
- Select a convenient multidimensional parametric latent distribution $P(Z)$ and train the model to make $N(Z|X)$ as close to $P(Z)$ as possible using Kullback-Lebler Divergence or $D_{KL}(N(Z|X) \| P(Z))$ as measure of approximation
- Combine $D_{KL}(N(Z|X) \| P(Z))$ with reconstruction loss measure: likelihood $\mathbb{E}_{N(Z|X)}[\ln Q(X|Z)]$ and maximize the combined measure

$$\mathbb{E}_{N(Z|X)}[\ln Q(X|Z)] - D_{KL}(N(Z|X) \| P(Z))$$

- First term motivates clustering codes, second term pushes clusters together. Compromise between the two is a continuous distribution on latent space

Kullback-Leibler Divergence for normal distributions

Let $p(x)$ and $q(x)$ be normal distribution densities

$$p(x)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad q(x)=\frac{1}{\sqrt{2\pi s^2}}e^{-\frac{(x-m)^2}{2s^2}}$$

$$\boxed{KL(p\|q)}=\int p(x)\ln\frac{p(x)}{q(x)}dx=\int p(x)\ln\frac{\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi s^2}}e^{-\frac{(x-m)^2}{2s^2}}}$$

$$=\int p(x)\ln\left(\frac{s^2}{\sigma^2}\right)^{\frac{1}{2}}dx+\int p(x)\left[\ln e^{-\frac{(x-\mu)^2}{2\sigma^2}}-\ln e^{-\frac{(x-m)^2}{2s^2}}\right]dx$$

$$=\int p(x)\ln\left(\frac{s^2}{\sigma^2}\right)^{\frac{1}{2}}dx+\int p(x)\left[-\frac{(x-\mu)^2}{2\sigma^2}+\frac{(x-m)^2}{2s^2}\right]dx=\frac{1}{2}\ln\frac{s^2}{\sigma^2}\overbrace{\int p(x)dx}^{=1}$$

$$-\frac{1}{2\sigma^2}\overbrace{\int(x-\mu)^2p(x)dx}^{=\sigma^2}+\frac{1}{2s^2}\int(x-m)^2p(x)dx=\frac{1}{2}\ln\frac{s^2}{\sigma^2}-\frac{1}{2}+\frac{1}{2s^2}\int(x-\mu+\mu-m)^2p(x)dx$$

$$=\frac{1}{2}\ln\frac{s^2}{\sigma^2}-\frac{1}{2}+\frac{1}{2s^2}\left[\overbrace{\int(x-\mu)^2p(x)dx}^{=\sigma^2}+(\mu-m)^2\overbrace{\int p(x)dx}^{=1}+2(\mu-m)\overbrace{\int(x-m)p(x)dx}^{=0}\right]$$

$$=\frac{1}{2}\ln\frac{s^2}{\sigma^2}-\frac{1}{2}+\frac{1}{2}\frac{\sigma^2}{s^2}+\frac{1}{2s^2}(\mu-m)^2=\boxed{\frac{1}{2}\left[\ln\frac{s^2}{\sigma^2}-1+\frac{\sigma^2+(\mu-m)^2}{s^2}\right]}$$

Generating New Samples with VAE

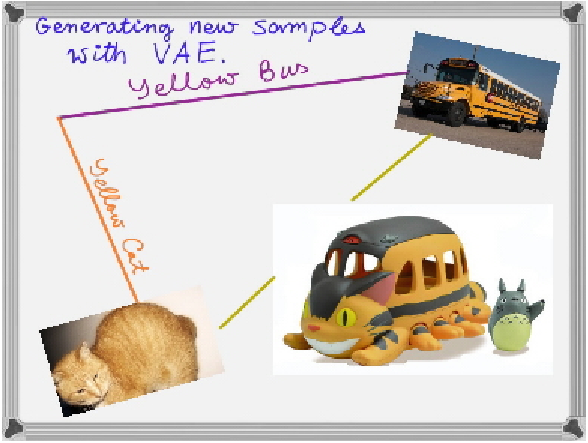


Figure: Sources: https://www.previewsworld.com/Catalog/MAY152491; weretable – reproduced under Creative Commons; H. Michael Miley reproduced under Creative Commons