Advanced Machine Learning
Summer 2019

Part 19 – Variational Autoencoders II
10.07.2019

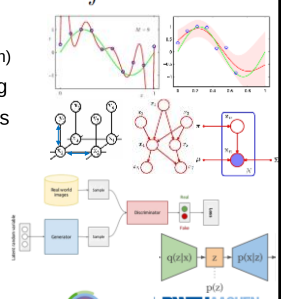
Prof. Dr. Bastian Leibe

RWTH Aachen University, Computer Vision Group
http://www.vision.rwth-aachen.de

Course Outline

- Regression Techniques
  - Linear Regression
  - Regularization (Ridge, Lasso)
  - Kernels (Kernel Ridge Regression)
- Deep Reinforcement Learning
- Probabilistic Graphical Models
  - Bayesian Networks
  - Markov Random Fields
  - Inference (exact & approximate)
  - Latent Variable Models
- Deep Generative Models
  - Generative Adversarial Networks
  - Variational Autoencoders

$f : \mathcal{X} \rightarrow \mathbb{R}$



Topics of This Lecture

- Recap: Variational Autoencoders
  - Autoencoders as Generative Models
  - Intractability
  - Variational Approximation
  - Evidence Lower Bound (ELBO)
- Applying VAEs
  - VAE Training
  - VAE Data Generation

Recap: Autoencoders

L_2 Loss function

Reconstructed input data $\hat{x}$

Decoder

Features z

Encoder

Input data x

Decoder: 4-layer upconv
Encoder: 4-layer conv

- How to learn such a feature representation?
  - Unsupervised learning approach for learning a lower-dimensional feature representation $\mathbf{z}$ from unlabeled input data $\mathbf{x}$.
  - $\mathbf{z}$ usually smaller than $\mathbf{x}$ (dimensionality reduction)
  - Want to capture meaningful factors of variation in the data Train such that features can be used to reconstruct original data.

Slide credit: Feifei Li

Recap: Autoencoders

Loss function (softmax, etc.)

Predicted label $\hat{y}$ y

Classifier

Features z

Encoder

Input data x

Fine-tune

bird plane
dog deer truck

Train for final task
(on small dataset)

- After training
  - Throw away the decoder part
  - Encoder can be used to initialize a supervised model
  - Fine-tune encoder jointly with supervised model
  - *Idea used in the 90s and early 2000s to pre-train deeper models*

Slide credit: Feifei Li

Recap: Variants of Autoencoders

L_2 Loss function

Reconstructed input data $\hat{x}$

Decoder

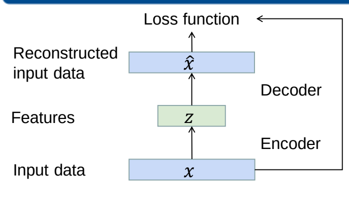
Features z

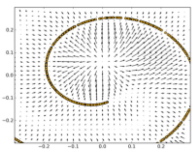
Encoder

Input data x

- Regularized Autoencoders
  - Include a regularization term to the loss function: $L\left(\mathbf{x}, g\left(f(\mathbf{x})\right)\right) + \Omega(\mathbf{z})$
  - E.g., enforce sparsity by an L_1 regularizer $\Omega(\mathbf{z}) = \lambda \sum_i |z_i|$

Recap: Variants of Autoencoders





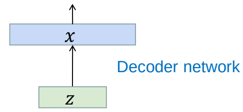
- Denoising Autoencoder (DAE)
  - Rather than the reconstruction loss, minimize $L\left(\mathbf{x}, g\left(f(\tilde{\mathbf{x}})\right)\right)$ where $\tilde{\mathbf{x}}$ is a copy of $\mathbf{x}$ that has been corrupted by some noise.
  - Denoising forces f and g to implicitly learn the structure of $p_{data}(\mathbf{x})$.

Image source: [Goodfellow 2016]

Recap: Probabilistic Spin on Autoencoders

Sample from true conditional $p_{\theta^*}(\mathbf{x}|\mathbf{z}^{(i)})$

Sample from true prior $p_{\theta^*}(\mathbf{z})$



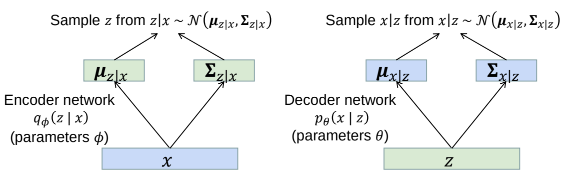
- Idea: Sample the model to generate data
  - We want to estimate the true parameters θ^* of this generative model.
- How should we represent the model?
  - Choose prior $p(\mathbf{z})$ to be simple, e.g., Gaussian
  - Conditional $p(\mathbf{x} \mid \mathbf{z})$ is complex (generates image)
    - ⇒ Represent with neural network
  - Learn model parameters to maximize likelihood of training data

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{z}) p_\theta(\mathbf{x} \mid \mathbf{z}) d\mathbf{z}$$ Intractable!

Slide adapted from Feifei Li

Recap: Variational Autoencoders

- Define additional encoder network $q_\phi(\mathbf{z} \mid \mathbf{x})$
  - Since we are modelling probabilistic generation of data, encoder and decoder networks are probabilistic



  - Encoder and decoder networks are also called recognition/inference and generation networks

D. Kingma, M. Welling, Auto-Encoding Variational Bayes, ICLR 2014

Slide credit: Feifei Li

Recap: Variational Autoencoders

- We can now work out the log-likelihood

Want to maximize data likelihood

$$\log p_\theta(x^{(i)}) = \mathbb{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ does not depend on } z)$$

$$= \mathbb{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z) p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)}$$

$$= \mathbb{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z) p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}$$

$$= \mathbb{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbb{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbb{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right]$$

$$= \boxed{\underbrace{\mathbb{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}\left(q_\phi(z \mid x^{(i)}) \| p_\theta(z)\right)}_{\mathcal{L}(x^{(i)}, \theta, \phi)}} + \underbrace{D_{KL}\left(q_\phi(z \mid x^{(i)}) \| p_\theta(z \mid x^{(i)})\right)}_{\geq 0}$$

Tractable lower bound, which we can take gradient of and optimize

Slide credit: Feifei Li

Recap: Variational Autoencoders

- Variational Lower Bound ("ELBO")

$$\log p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)}, \theta, \phi)$$

$$= \underbrace{\mathbb{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right]}_{\text{"Reconstruct the input data"}} - \underbrace{D_{KL}\left(q_\phi(z \mid x^{(i)}) \| p_\theta(z)\right)}_{\text{"Make approximate posterior distribution close to prior"}}$$

- Training: Maximize lower bound

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)}, \theta, \phi)$$

Slide adapted from Feifei Li

Topics of This Lecture

- Recap: Variational Autoencoders
  - Autoencoders as Generative Models
  - Intractability
  - Variational Approximation
  - Evidence Lower Bound (ELBO)
- Applying VAEs
  - VAE Training
  - VAE Data Generation

Applying Variational Autoencoders

- Putting it all together…
  - Maximizing the likelihood lower bound

$$\underbrace{\mathbb{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}\left(q_\phi(z \mid x^{(i)}) \| p_\theta(z)\right)}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

  - Let's look at computing the bound for a given minibatch of input data (forward pass)…

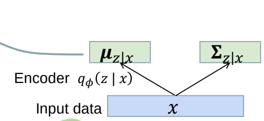
Input data x

Applying Variational Autoencoders

- Putting it all together…
  - Maximizing the likelihood lower bound

$$\underbrace{\mathbb{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}\left(q_\phi(z \mid x^{(i)}) \| p_\theta(z)\right)}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

Make approximate posterior distribution close to prior

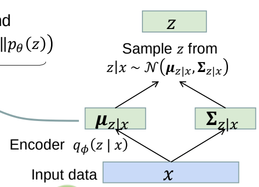


Applying Variational Autoencoders

- Putting it all together…
  - Maximizing the likelihood lower bound

$$\underbrace{\mathbb{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}\left(q_\phi(z \mid x^{(i)}) \| p_\theta(z)\right)}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

Make approximate posterior distribution close to prior

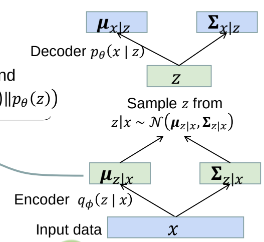


Applying Variational Autoencoders

- Putting it all together…
  - Maximizing the likelihood lower bound

$$\underbrace{\mathbb{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}\left(q_\phi(z \mid x^{(i)}) \| p_\theta(z)\right)}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

Make approximate posterior distribution close to prior



Applying Variational Autoencoders

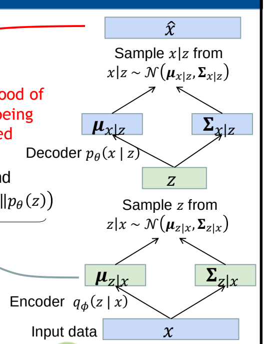
- Putting it all together…

Maximize likelihood of original input being reconstructed

  - Maximizing the likelihood lower bound

$$\underbrace{\mathbb{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}\left(q_\phi(z \mid x^{(i)}) \| p_\theta(z)\right)}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

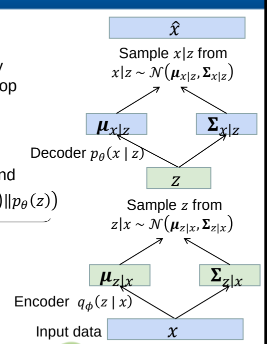
Make approximate posterior distribution close to prior



Applying Variational Autoencoders

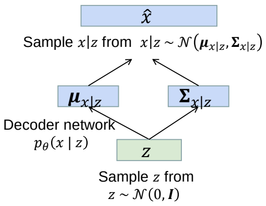
- Putting it all together…
  - Compute this forward pass for every minibatch of input data, then backprop
  - Maximizing the likelihood lower bound

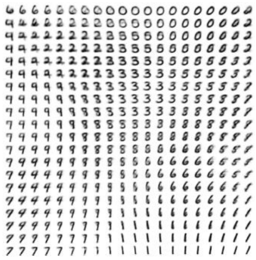
$$\underbrace{\mathbb{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}\left(q_\phi(z \mid x^{(i)}) \| p_\theta(z)\right)}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$



Variational Autoencoders: Generating Data

- Use decoder network
  - Now sample z from prior





Latent MNIST manifold

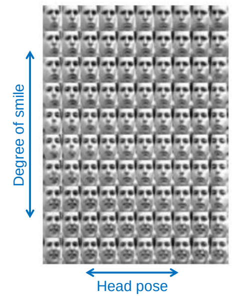
D. Kingma, M. Welling, Auto-Encoding Variational Bayes, ICLR 2014

Slide credit: Feifei Li

Image source: [Kingma 2013]

Variational Autoencoders: Generating Data

- Another example
  - Learning a face manifold
- Comments
  - Diagonal prior on $\mathbf{z}$
  - $\Rightarrow$ Independent latent variables
  - Different dimensions of $\mathbf{z}$ encode interpretable factors of variation



Slide adapted from Feifei Li

Image source: [Kingma 2013]

Some More Learned Manifolds

32x32 CIFAR-10

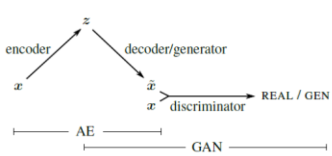
Labeled Faces in the Wild

Image source: [Kingma 2016], [Larsen 2017]

Summary: Variational Autoencoders

- Idea
  - Probabilistic Spin on traditional autoencoders
  - Intractable density $\Rightarrow$ derive & optimize a variational lower bound
- Pros
  - Principled approach to generative models
  - Allows inference of $q_\phi(z \mid x)$, can be useful feature representation for other tasks
- Cons
  - Only maximizes lower bound of likelihood
  - Samples blurrier and lower quality compared to state-of-the-art (GANs)
- Active area of research
  - More flexible approximations, e.g., GMMs instead of diagonal Gaussian

Combinations



- Attempts at combining the advantages
  - Use learned feature representations in the GAN discriminator as basis for the VAE reconstruction objective
  - Replacing element-wise errors with feature-wise errors to better capture the data distribution

A. Larsen, S. Sonderby, H. Larochelle, O. Winther, Autoencoding beyond Pixels using a Learned Similarity Metric, arXiv 1512.09300

Image source: [Larsen 2015]

Results

Samples from different generative models

Reconstructions from different autoencoders

- $\text{VAE}_{\text{Dis}_l}$: Train a GAN first, then use the discriminator to train a VAE
- VAE/GAN: VAE and GAN trained together

Image source: [Larsen 2015]

References

- Variational Auto-Encoders
  - D. Kingma, M. Welling, Auto-Encoding Variational Bayes, ICLR 2014.
  - A. Larsen, S. Sonderby, H. Larochelle, O. Winther, Autoencoding beyond Pixels using a Learned Similarity Metric, arXiv:1512.09300, 2015.