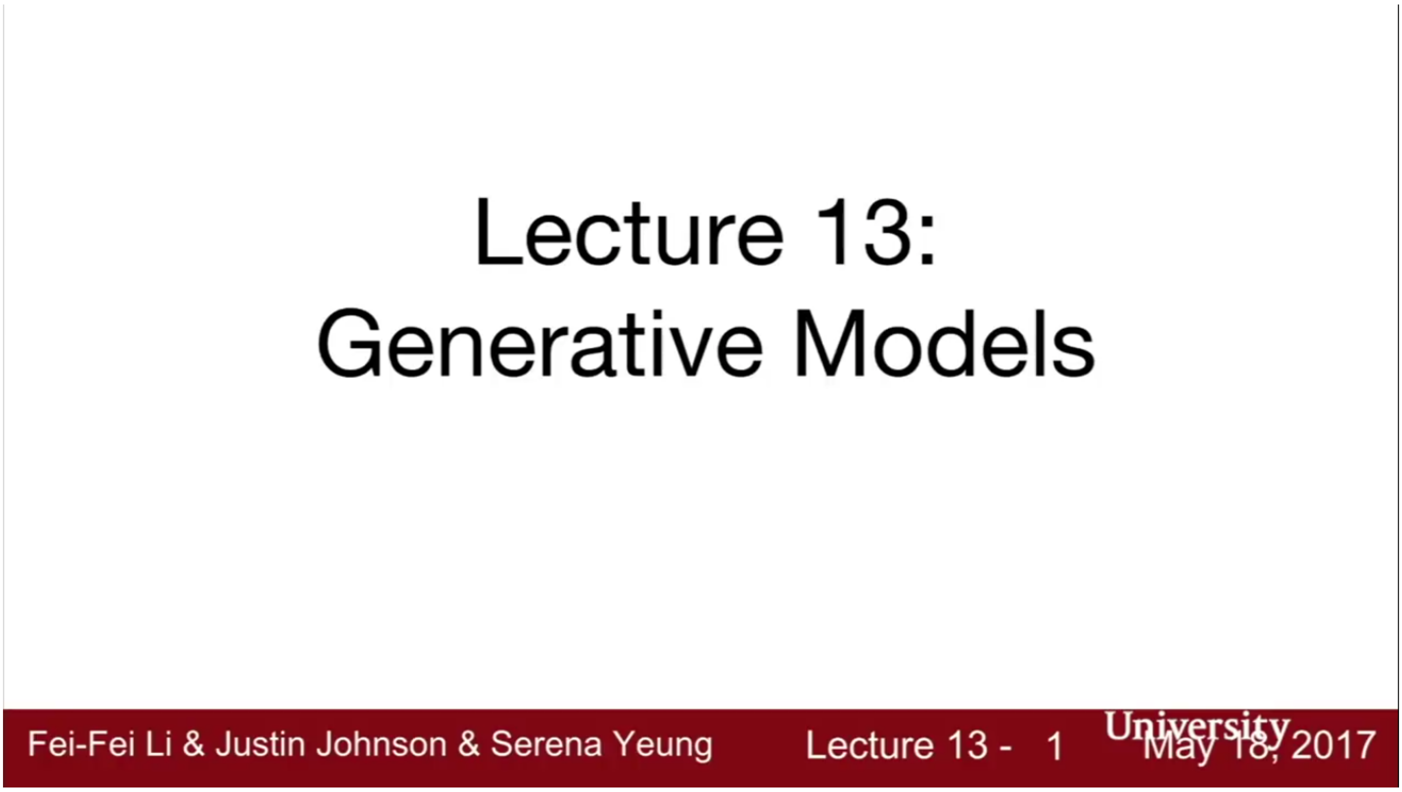
lec13

Creation Date: 24/01/2020 18:12

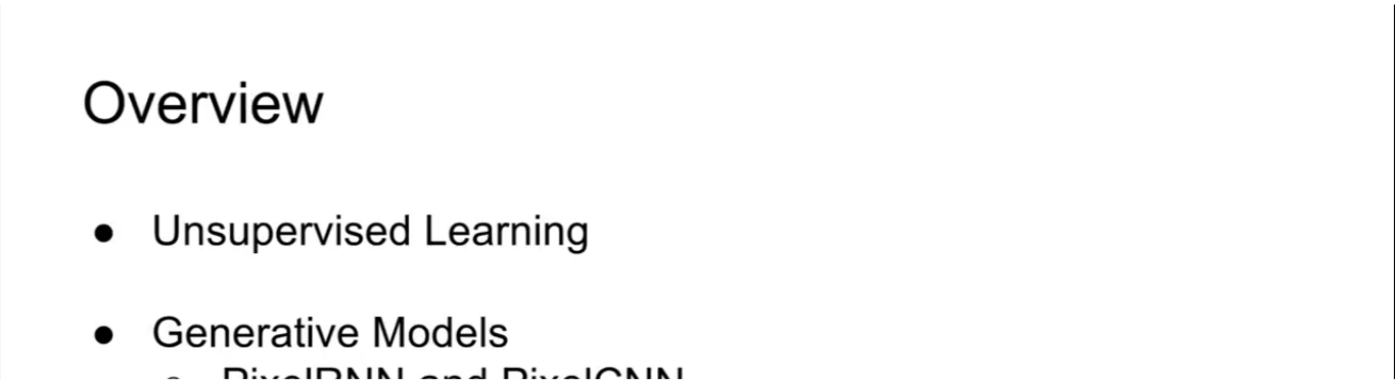
Last Modified Date: 27/01/2020 17:19

Lec 13 : Generative Models



Lecture 13:
Generative Models

Fei-Fei Li & Justin Johnson & Serena Yeung Lecture 13 - 1 University May 18, 2017



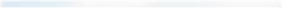
Overview

- Unsupervised Learning
- Generative Models
  - PixelRNN and PixelCNN

- Generative Models: It is a type of Unsupervised Learning.

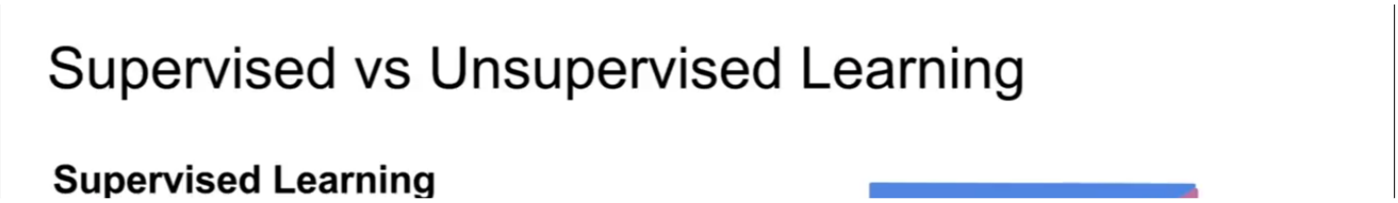
Supervised vs Unsupervised Learning

Supervised Learning



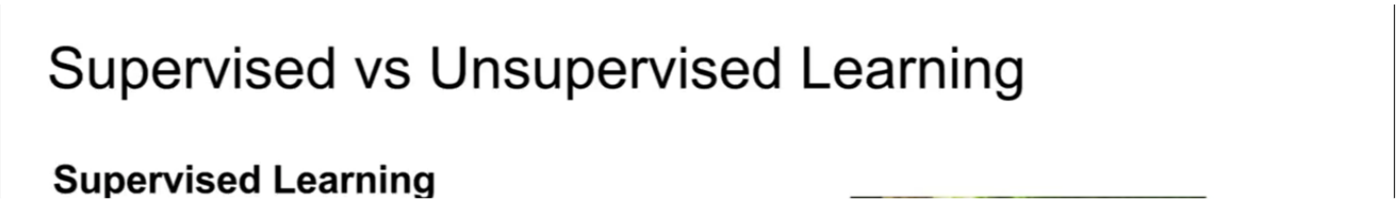
Supervised vs Unsupervised Learning

Supervised Learning



Supervised vs Unsupervised Learning

Supervised Learning



Supervised vs Unsupervised Learning

Supervised Learning

- Goal of Unsupervised Learning: To learn the underlying structure of the given data.

Supervised vs Unsupervised Learning

Supervised vs Unsupervised Learning

Supervised vs Unsupervised Learning

Supervised vs Unsupervised Learning

- Density Estimation: Modeling the data such that the probability of the data generation is high where more points are concentrated.

- Differences between Supervised and Unsupervised Learning:

Supervised vs Unsupervised Learning

- Solving Unsupervised Learning problems helps in understanding the structure of the visual world.

Generative Models

- **Generative models address density estimation.**
- **Explicit Density Estimation**: Explicitly define and solve for $p_{model}(x)$
- **Implicit Density Estimation**: We learn a model that can sample from $p_{model}(x)$ ***without*** explicitly defining it.

Why Generative Models?

Taxonomy of Generative Models

Direct

PixelRNN and PixelCNN:

PixelRNN and PixelCNN

Fully visible belief network

PixelRNN *[van der Oord et al. 2016]*

Generate image pixels starting from corner

PixelCNN *[van der Oord et al. 2016]*

Softmax loss at each pixel

PixelCNN *[van der Oord et al. 2016]*

- The generation depends on the value of the starting pixel, because everything else is conditioned on that initial pixel.

Generation Samples

PixelRNN and PixelCNN

Variational Autoencoders

Variational Autoencoders (VAE)

So far...

PixelCNNs define tractable density function, optimize likelihood of training data:

Some background first: Autoencoders

- Ans: We want dimensionality reduction since we want a compact representation of x but also which represent all the important features of x.

Some background first: Autoencoders

Some background first: Autoencoders

Reconstructed data

Some background first: Autoencoders

Reconstructed data

Some background first: Autoencoders

Some background first: Autoencoders

- ***So using large amount of unlabeled data we can learn a better feature representation and then use it to initialize a supervised learning problem where we only have small data.***
- This approach helps because with small amount of data supervised approaches will cause overfitting.
- So using this approach we get a better initilization of the supervised model with better features.
- ***Features capture factors of variation in training data.***

Some background first: Autoencoders

Variational Autoencoders

Variational Autoencoders

- Variational Autoencoders help us in sampling
- The intuition is that each element $z^{(i)}$ captures certain feature of x (like smile, eye color of faces).
- z is modeling using some priors, more commonly we use Gaussian priors (μ, σ).

Variational Autoencoders

We want to estimate the true parameters $\theta*$ of this generative model

- We choose the prior $p(z)$ to simple, like Gaussian. Because it is reasonable for latent attributes.
- The conditional $p(x|z)$ is complex in nature since it is generating images where $p(z)$ is just a vector so we try to model z using Gaussian.

Variational Autoencoders

We want to estimate the true parameters $\theta*$ of this generative model

Variational Autoencoders

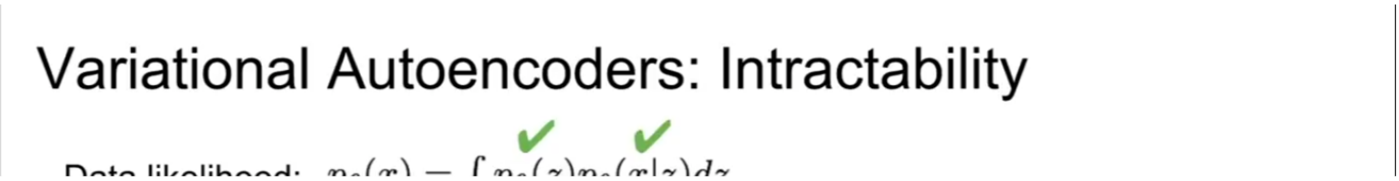
We want to estimate the true parameters $\theta*$ of this generative model

Variational Autoencoders

We want to estimate the true parameters θ^* of this generative model

Variational Autoencoders: Intractability

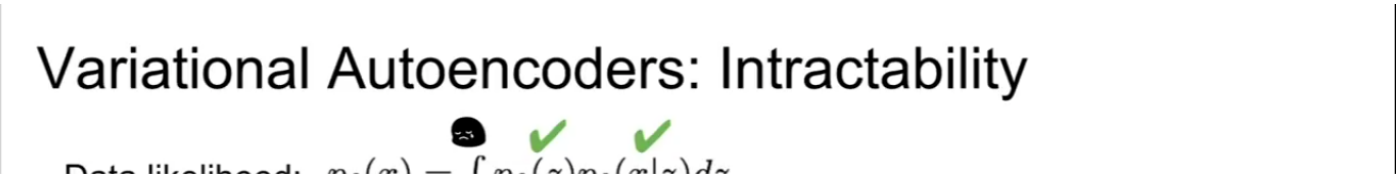
Data likelihood: $p_\theta(x) = \int \overset{\checkmark}{p_\theta(z)} p_\theta(x|z) dz$



Variational Autoencoders: Intractability

✓ ✓

Data likelihood: $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$



Variational Autoencoders: Intractability

✓ ✓

Data likelihood: $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$

- The integral is intractable.

Variational Autoencoders: Intractability

😞 ✔ ✔

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Variational Autoencoders: Intractability

😞 ✔ ✔

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Variational Autoencoders

Since we’re modeling probabilistic generation of data, encoder and decoder networks are probabilistic

Variational Autoencoders

Since we’re modeling probabilistic generation of data, encoder and decoder networks are probabilistic

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

Variational Autoencoders

Putting it all together: maximizing the

Variational Autoencoders

Putting it all together: maximizing the

Variational Autoencoders

Putting it all together: maximizing the

Variational Autoencoders

Putting it all together: maximizing the

Variational Autoencoders

Putting it all together: maximizing the

Variational Autoencoders

Putting it all together: maximizing the

Variational Autoencoders: Generating Data!

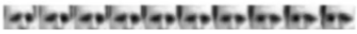
Use decoder network. Now sample z from prior!

Variational Autoencoders: Generating Data!

Use decoder network. Now sample z from prior!

Data manifold for 2-d z

Variational Autoencoders: Generating Data!



Variational Autoencoders: Generating Data!

Variational Autoencoders

Probabilistic spin to traditional autoencoders => allows generating data

GANs:

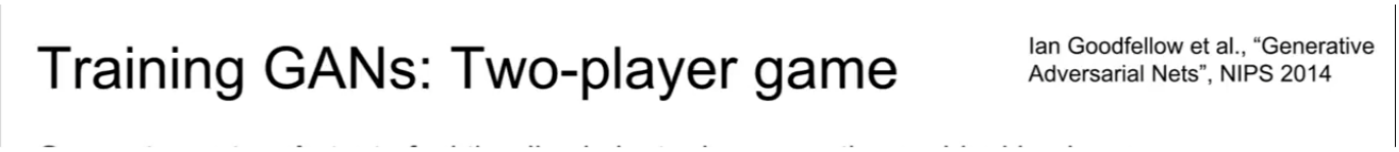
So far...

PixelCNNs define tractable density function, optimize likelihood of training data:

$$p_\theta(x) = \prod^{n} p_\theta(x_i|x_1, ..., x_{i-1})$$

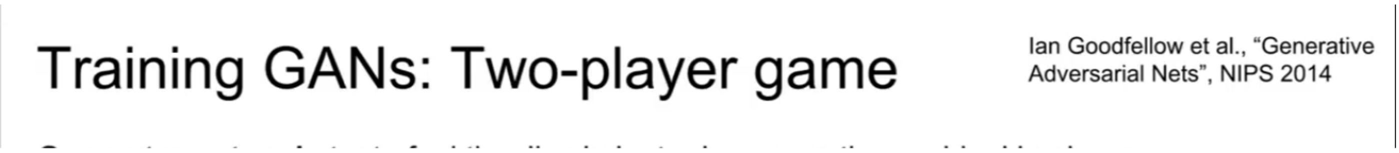
Generative Adversarial Networks

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014



Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014



Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Minimax objective function:

Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Minimax objective function:

Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Generative Adversarial Nets

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Generated samples

Generative Adversarial Nets

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

Generated samples (CIFAR-10)

Generative Adversarial Nets: Convolutional Architectures

Generator is an upsampling network with fractionally-strided convolutions

Generative Adversarial Nets: Convolutional Architectures

Generative Adversarial Nets: Convolutional Architectures

Generative Adversarial Nets: Convolutional Architectures

Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016

Generative Adversarial Nets: Interpretable Vector Math

Glasses man No glasses man No glasses woman

Radford et al

2017: Year of the GAN

Better training and generation Source->Target domain transfer

Text -> Image Synthesis

this small bird has a pink breast and crown, and black

this magnificent fellow is almost all black with a red

“The GAN Zoo”

See also: https://github.com/soumith/ganhacks for tips and tricks for trainings GANs

GANs

Don’t work with an explicit density function
Take game-theoretic approach: learn to generate from training distribution through 2-player

Recap

Generative Models

- PixelRNN and PixelCNN Explicit density model, optimizes exact likelihood, good