Outline

- Generative Models Basics
- **Autoregressive Models**
- Autoencoder and Variational Autoencoder
- Generative Adversarial Network
- Diffusion Models

Fully visible belief network

Explicit density model

Use chain rule to decompose likelihood of an image x into product of 1-d distributions:

$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, ..., x_{i-1})$$

Likelihood of image x

Probability of i'th pixel value given all previous pixels

Then maximize likelihood of training data

Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Fully visible belief network

Explicit density model

Use chain rule to decompose likelihood of an image x into product of 1-d distributions:

$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, ..., x_{i-1})$$

Likelihood of image x

Probability of i'th pixel value given all previous pixels

Then maximize likelihood of training data

Complex distribution over pixel values => Express using a neural network!

Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Fully visible belief network

Explicit density model

Use chain rule to decompose likelihood of an image x into product of 1-d distributions:

$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, ..., x_{i-1})$$

Likelihood of image x

Probability of i'th pixel value given all previous pixels

Will need to define ordering of "previous pixels"

Complex distribution over pixel values => Express using a neural network!

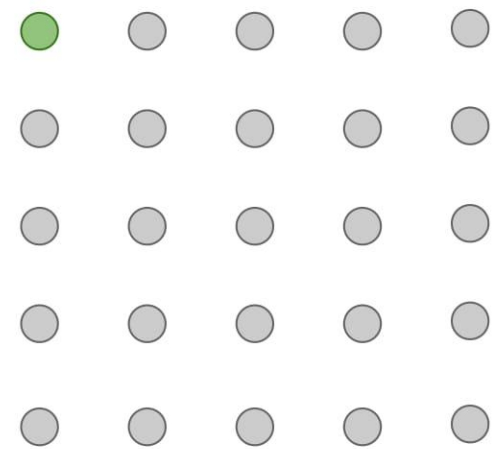
Then maximize likelihood of training data

Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

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

Generate image pixels starting from corner

Dependency on previous pixels modeled using an RNN (LSTM)

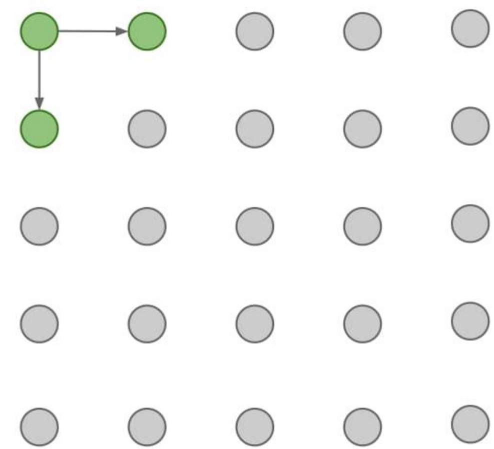


Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

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

Generate image pixels starting from corner

Dependency on previous pixels modeled using an RNN (LSTM)

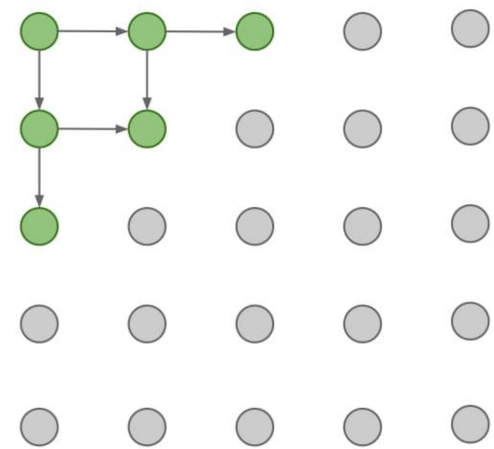


Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

PixelRNN [van der Oord et al. 2016]

Generate image pixels starting from corner

Dependency on previous pixels modeled using an RNN (LSTM)

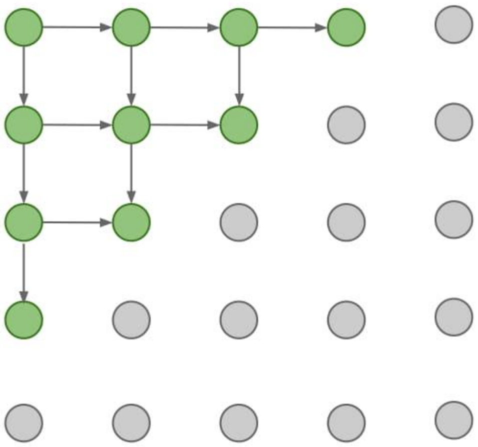


Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

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

Generate image pixels starting from corner

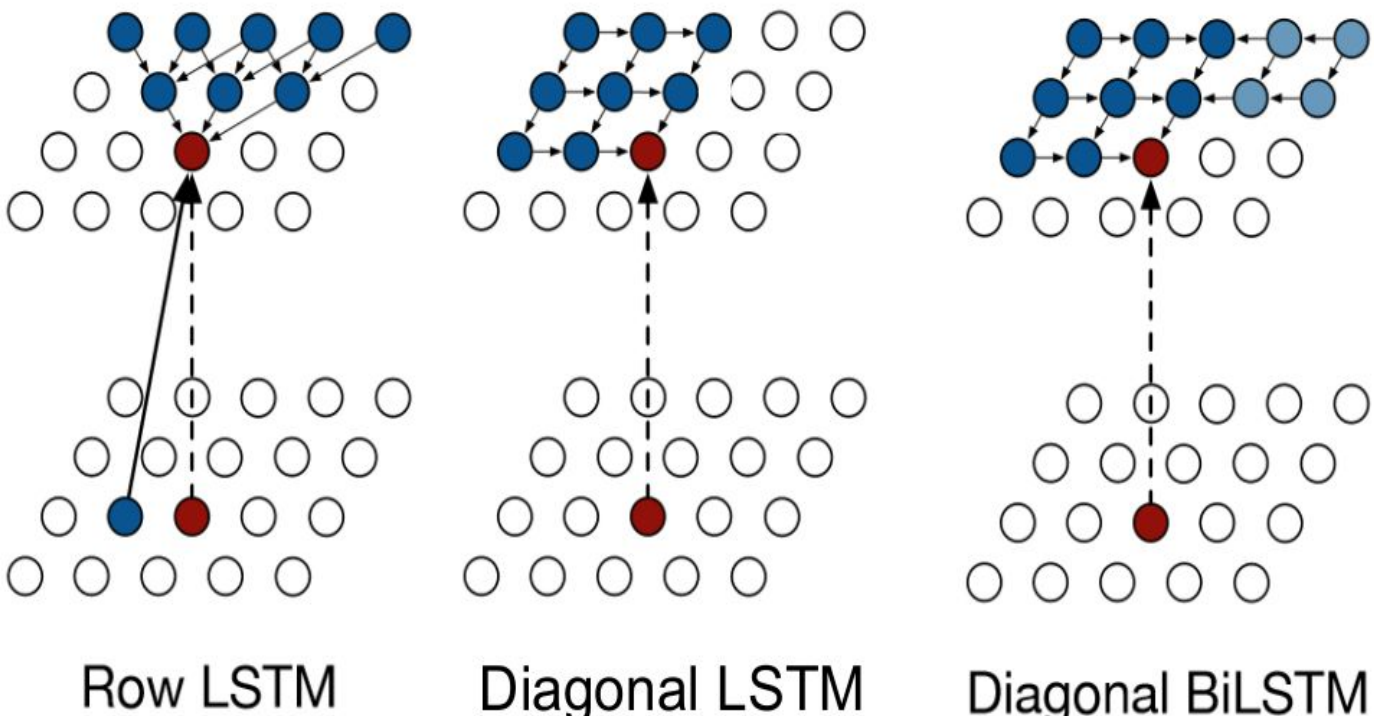
Dependency on previous pixels modeled using an RNN (LSTM)



Drawback: sequential generation is slow!

Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variants of Pixel RNN (LSTM)



Pixel recurrent neural networks, ICML 2016

Slide credit: Yohei Sugawara

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

Still generate image pixels starting from corner

Dependency on previous pixels now modeled using a CNN over context region

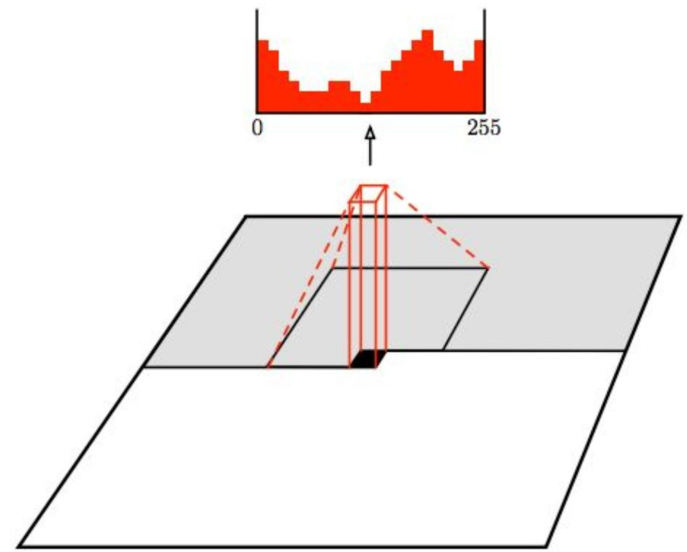


Figure copyright van der Oord et al., 2016. Reproduced with permission.

Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

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

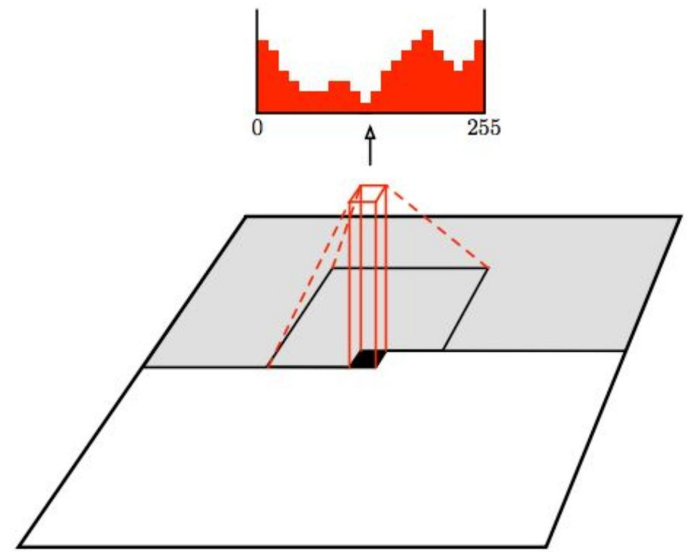
Still generate image pixels starting from corner

Dependency on previous pixels now modeled using a CNN over context region

Training: maximize likelihood of training images

$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, ..., x_{i-1})$$

Softmax loss at each pixel



Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

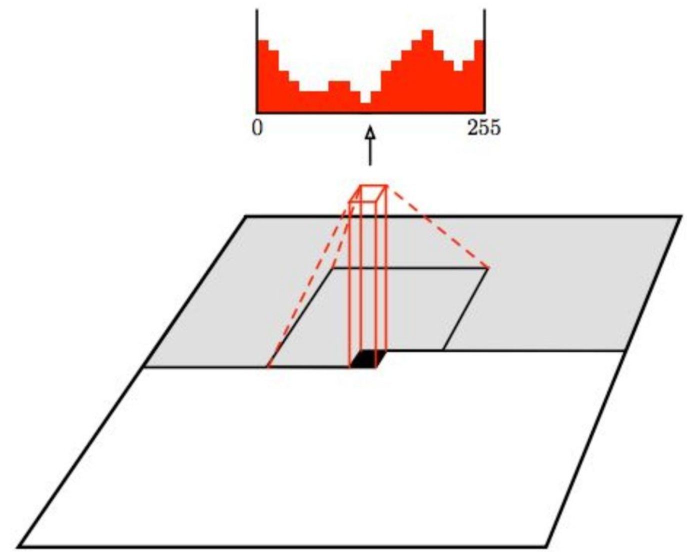
PixelCNN *[van der Oord et al. 2016]*

Still generate image pixels starting from corner

Dependency on previous pixels now modeled using a CNN over context region

Training is faster than PixelRNN
(can parallelize convolutions since context region values known from training images)

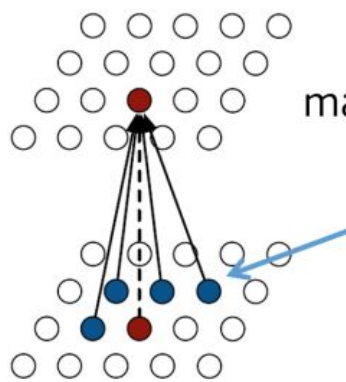
Generation must still proceed sequentially
=> still slow



Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

PixelCNN

- 2D convolution on previous layer
- Apply masks so a pixel does not see future pixels (in sequential order)

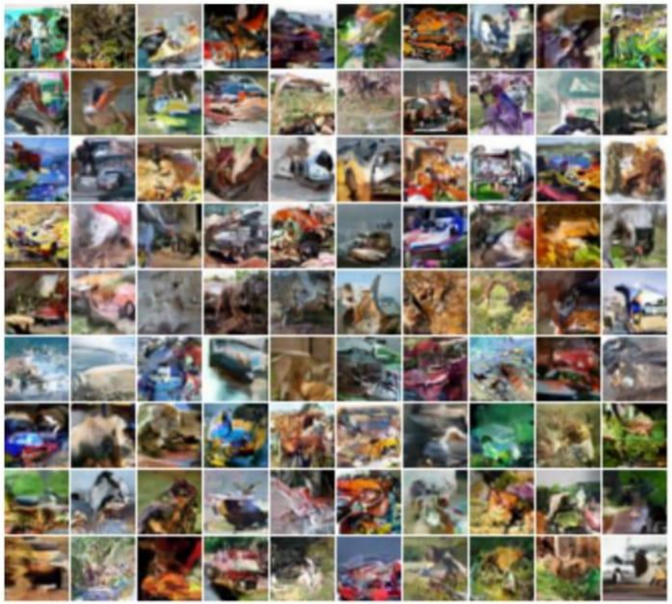


1	1	1
1	0	0
0	0	0

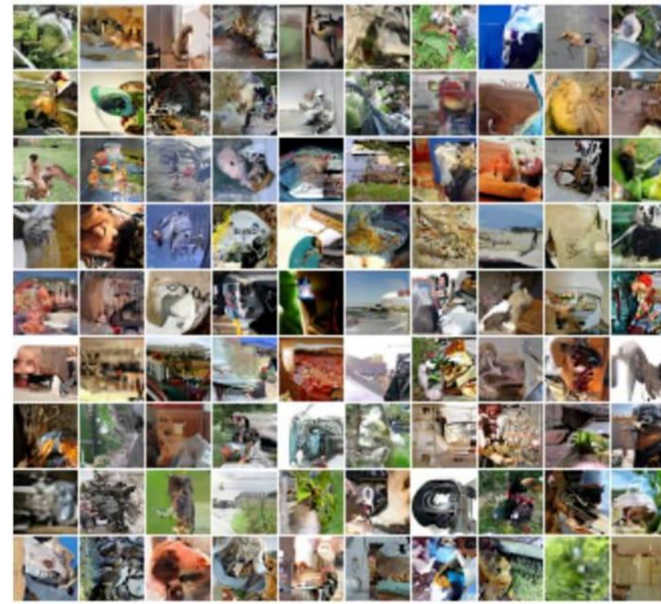
Pixel recurrent neural networks, ICML 2016

Slide credit: Yohei Sugawara

Generation Samples



32x32 CIFAR-10



32x32 ImageNet

Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

PixelRNN and PixelCNN

Pros:
- Can explicitly compute likelihood p(x)
- Explicit likelihood of training data gives good evaluation metric
- Good samples

Con:
- Sequential generation => slow

Improving PixelCNN performance
- Gated convolutional layers
- Short-cut connections
- Discretized logistic loss
- Multi-scale
- Training tricks
- Etc…

See
- Van der Oord et al. NIPS 2016
- Salimans et al. 2017 (PixelCNN++)

Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung