PixelCNN Models with Auxiliary Variables for Natural Image Modeling

Alexander Kolesnikov, IST Austria (joint work with Christoph Lampert)

The Third Christmas Colloquium on Computer Vision, Moscow, Russia

Introduction

$q(X)$ – true probability density over natural image $X \in \mathcal{X}$.

$X = \{x_1, \ldots, x_n\}$ – collection of n pixels; $x_i \in \{0, \ldots, 255\}$

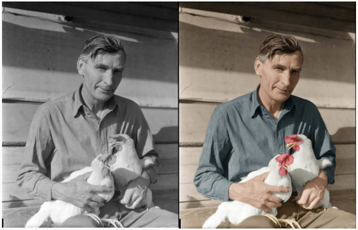
$D = \{X_1, X_2, \ldots, X_m\}$ – i.i.d. samples from q.

Goal: use D to estimate probability density $p(X) = p(x_1, \ldots, x_n)$, which is close to q.

$p(X)$ is also called *likelihood of* X.

- $p(X) \geq 0$
- $\sum_{X \in \mathcal{X}} p(X) = 1$
- Min. $\mathrm{KL}(q||p) \Leftrightarrow$ Max. $\mathbb{E}_{X \sim q} \log p(X) \approx \frac{1}{n} \sum_{X \in D} \log p(X)$

Motivation: image manipulations



Automatic colorization

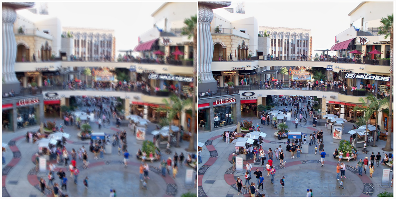


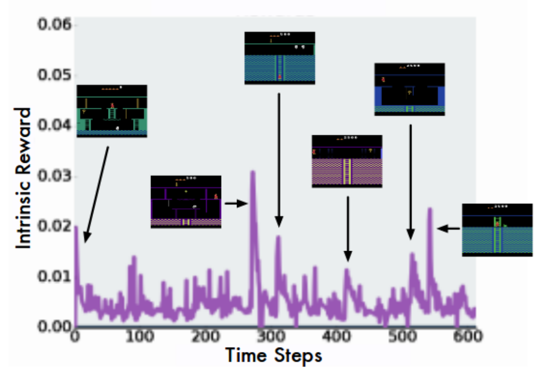
Image debluring

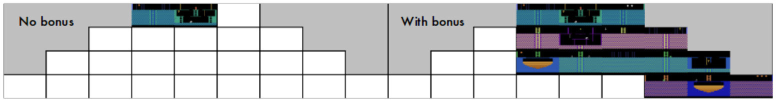
X' – corrupted image

$$X^* = \underset{X \in \mathcal{X}}{\operatorname{argmax}}\, p(X) + \mathrm{similarity}(X, X').$$

Motivation: reinforcement learning

Count-Based Exploration with Neural Density Models by *Ostrovskiy et al.*

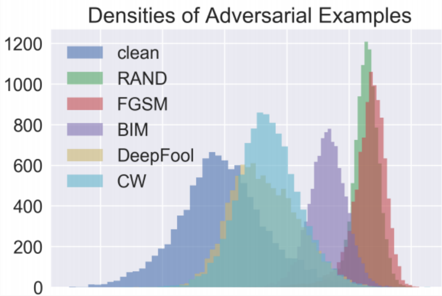




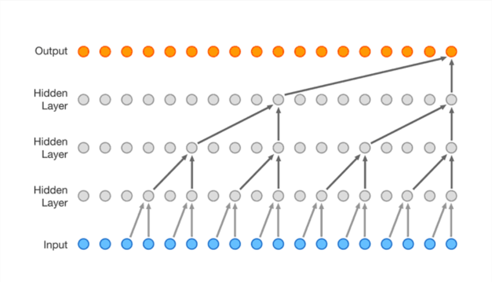
Motivation: Defense against Adversarial Samples

PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples by Song et al.

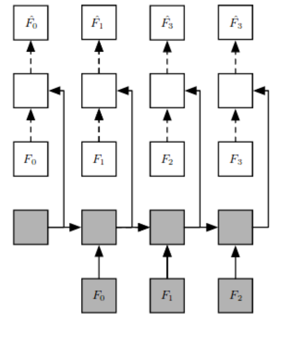
Category: **wi-fi router**



Motivation: generalization to other domains



Wavenet for modeling audio



Video Pixel Network

Image modeling techniques **generalize** beyond images:

- Audio [Oord, Dieleman, et al. 2016]
- Video [Kalchbrenner et al. 2016]
- Natural language [Gulrajani et al. 2017]

Research landscape

GANs: Generative Adversarial Networks

- Implicit likelihood. Learns generator $G : \mathcal{Z} \to \mathcal{X}$

VAEs: Variational AutoEncoders

- Intractable latent variable model: $p(X) = \int_z p(X|z)p(z)dz$

Tractable models

- Explicit and computationally tractable likelihood $p(X)$

Research landscape

GANs: Generative Adversarial Networks

- Implicit likelihood. Learns generator $G : \mathcal{Z} \rightarrow \mathcal{X}$

VAEs: Variational AutoEncoders

- Intractable latent variable model: $p(X) = \int_z p(X|z)p(z)dz$

Tractable models

- Explicit and computationally tractable likelihood $p(X)$
  Autoregressive model + conv. network = PixelCNN

Main result

Introduce autoregressive models with auxiliary variables:

- Improved perceptual quality of produced samples.
- Improved sampling speed.

This talk

Background: PixelCNN model

[Oord el al, ICML 2016]

[Oord el al, NIPS 2016]

PixelCNN Model

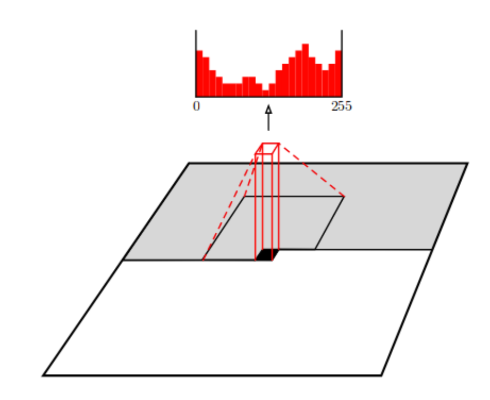
Key modeling idea: elementary chain rule

Represent a distribution over natural images as a product of one-dimensional conditional distributions:

$$p(X) = p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_1, \ldots, x_{i-1})$$

- Need to model only **1-dimensional** distributions!
- Images are translation invariant, thus **all conditional distributions can be modeled by a single function**
- **All conditional distributions** can be computed in a single forward pass of an appropriate convolutional network

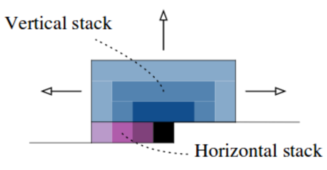
PixelCNN model

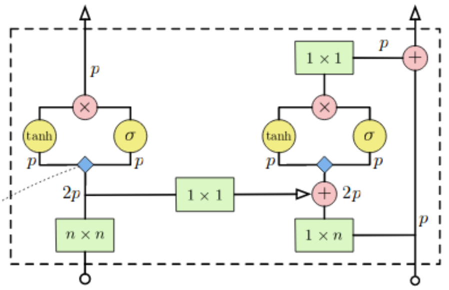


An illustration of how a single conditional distribution is computed.

Image credit: Conditional Image Generation with PixelCNN Decoders by Oord et al.

PixelCNN: implementation details





- Displaced receptive field
- Gatinig non-linearity: $\phi(x) = \tanh(x_l) * \sigma(x_r)$

Straightforward to derive conditional model: $p(X|z)$

Image credit: Conditional Image Generation with PixelCNN Decoders by Oord et al.

PixelCNN: Training

$D = \{X_1, X_2, \dots, X_m\}$ – training data

w – model parameters

Training procedure

Log-likelihood maximization using stochastic gradient descent:

$$\max_w \frac{1}{m} \sum_{X \in D} \sum_{i=1}^{n} \log p_w(x_i | x_1, \dots, x_{i-1})$$

Gradients can be computed in a single backward pass

PixelCNN: Sampling

Sampling is sequential:

$x_1 \sim p(x_1)$

$x_2 \sim p(x_2|x_1)$

$x_3 \sim p(x_3|x_2, x_1)$

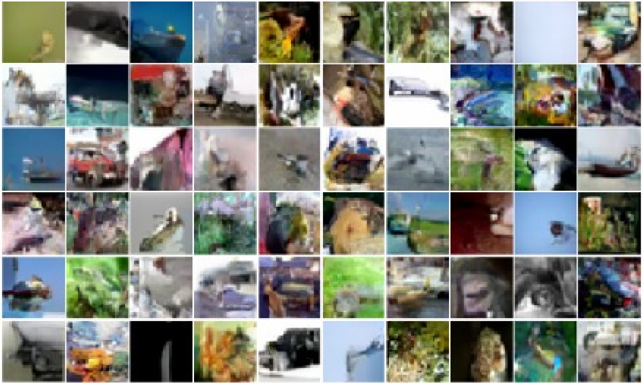
$x_4 \sim p(x_4|x_3, x_2, x_1)$

$\ldots$

Image credit: https://github.com/PrajitR/fast-pixel-cnn

Shortcomings of the PixelCNN model

- **Lack of global structure** in image samples:



- **Slow sampling**: deep neural networks needs to be invoked for every pixel which is being generated.

PixelCNN models with Auxiliary Variables (ICML 2017)

PixelCNN with Auxiliary Variables

We extend PixelCNN with auxiliary variable, $\widehat{X}$ and model a **joint** probability distribution:

$$p(X, \widehat{X}) = p(X|\widehat{X})p(\widehat{X})$$

Crucial modeling assumption:

$\widehat{X} = f(X)$, where f is a deterministic function.

In this talk: $\widehat{X} = f(X)$ is an image, also modeled by PixelCNN.

Efficient training: $\max \sum_{X \in \mathcal{D}} [\log p(f(X)) + \log p(X|f(X))]$

Efficient sampling: sample $\widehat{X}$ from $p(\widehat{X})$, then X from $p(X|\widehat{X})$

Grayscale PixelCNN

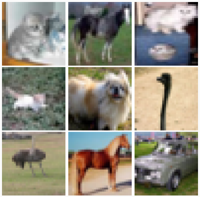
Problem: Lack of global structure in random image samples

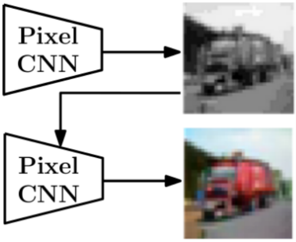
Solution: Explicitly encourage model to capture global image statistics

$\widehat{X}$ – 4-bit quantized grayscale view of X

4-bit grayscale

original





$\widehat{X}$ retains global image information and omits distracting texture patterns

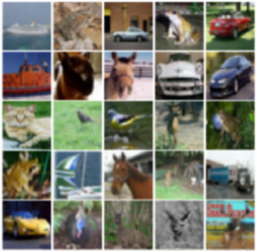
Grayscale PixelCNN: insight into image modeling problem

Average loss of the trained model:

$$-\log p(\widehat{X}) \approx \mathbf{0.46} \qquad\qquad -\log p(X|\widehat{X}) \approx 2.52$$

Likelihood objective is dominated by colors/texture patterns

original

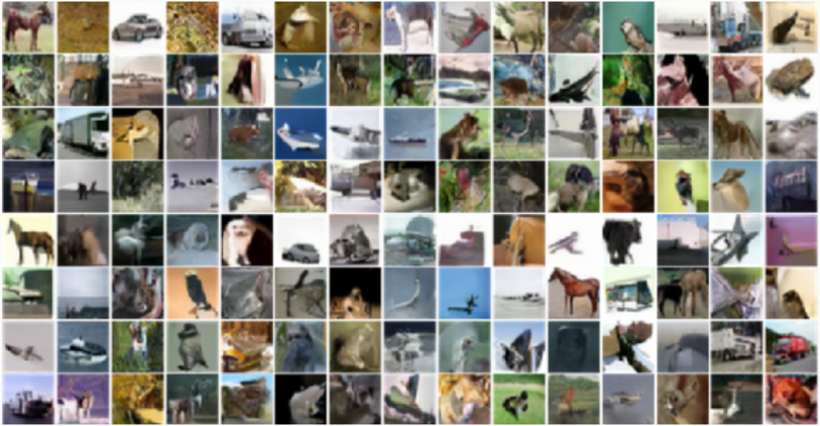


4-bit grayscale

samples from $p(X|\widehat{X})$

4-bit grayscale image transform is sufficient to produce photo-realistic images

Grayscale PixelCNN: samples



Grayscale PixelCNN samples are globally coherent.

Grayscale PixelCNN: likelihood

Model	Bits per dim.
Deep Diffusion [Sohl-Dickstein et al. 2015]	≤ 5.40
NICE [Dinh, Krueger, and Y. Bengio 2014]	4.48
DRAW [Gregor, Danihelka, et al. 2015]	≤ 4.13
Deep GMMs [Oord and Schrauwen 2014]	4.00
Conv Draw [Gregor, Besse, et al. 2016]	≤ 3.58
Real NVP [Dinh, Sohl-Dickstein, and S. Bengio 2016]	3.49
Matnet + AR [Bachman 2016]	≤ 3.24
PixelCNN [Oord, Kalchbrenner, and Kavukcuoglu 2016]	3.14
VAE with IAF [Kingma, Salimans, and Welling 2016]	≤ 3.11
Gated PixelCNN [Oord, Kalchbrenner, Espeholt, et al. 2016]	3.03
PixelRNN [Oord, Kalchbrenner, and Kavukcuoglu 2016]	3.00
Grayscale PixelCNN [this talk]	≤ 2.98
DenseNet VLAE [Chen et al. 2017]	≤ 2.95
PixelCNN++ [Salimans et al. 2017]	2.92

The negative log-likelihood of the different models for the CIFAR-10 **test set** measured as bits-per-dimension.

Pyramid PixelCNN

Problem: Slow sampling

Solution: Multiscale decomposition

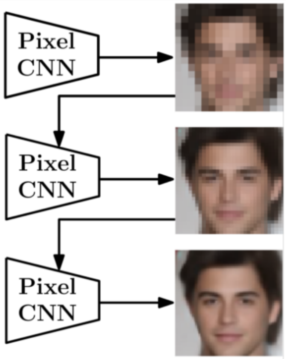
$\widehat{X}$ – a low-resolution view of X.

Can be applied recursively.

Start from generating tiny 8x8 seed

Upscale 4 times to 128x128

Image generation = super-resolution



Tiny network for each step:

3 residual blocks $\Longrightarrow$ fast sampling

Pyramid PixelCNN

Problem: Slow sampling

Solution: Multiscale decomposition

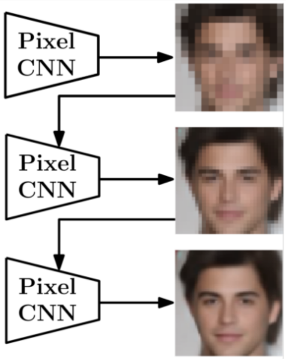
$\widehat{X}$ – a low-resolution view of X.

Can be applied recursively.

Start from generating tiny 8x8 seed

Upscale 4 times to 128x128

Image generation = super-resolution



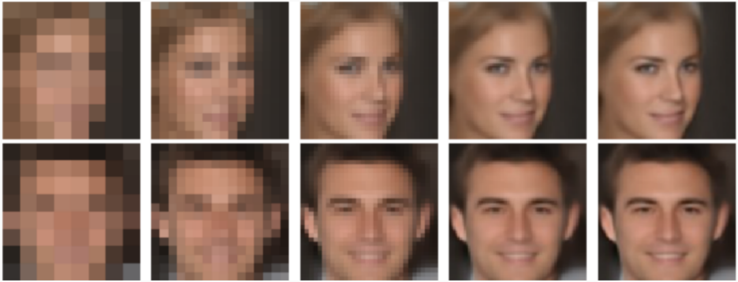
Tiny network for each step:

3 residual blocks $\implies$ fast sampling

Concurrent work: Parallel autoregressive density estimation

Imporoves complexity by Reed et al: $\mathcal{O}(n) \to \mathcal{O}(\log n)$

Pyramid PixelCNN: samples



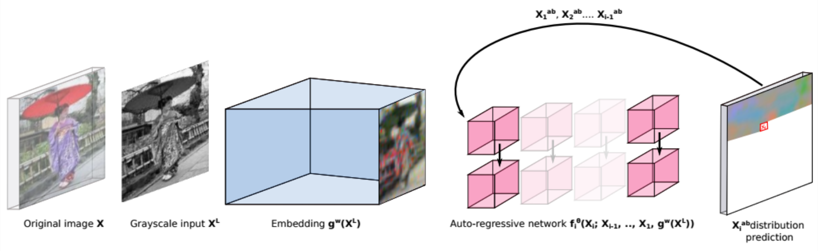
Despite tiny PixelCNN models, which are used, high-resolution samples are accurate and globally coherent samples.

Summary

- High-frequency texture patterns dominate the likelihood objective
- Auxiliary quantized grayscale variable can be used to encourage PixelCNN model to focus more on semantic structure of an image
- Low-resolution auxiliary variables help to scale PixelCNN for high-resolution images
- Grayscale and low-resolution auxiliary variables can be combined

Application: Probabilistic Image Colorization

Joint work with Amèlie Royer and Christoph Lampert (BMVC 2017)

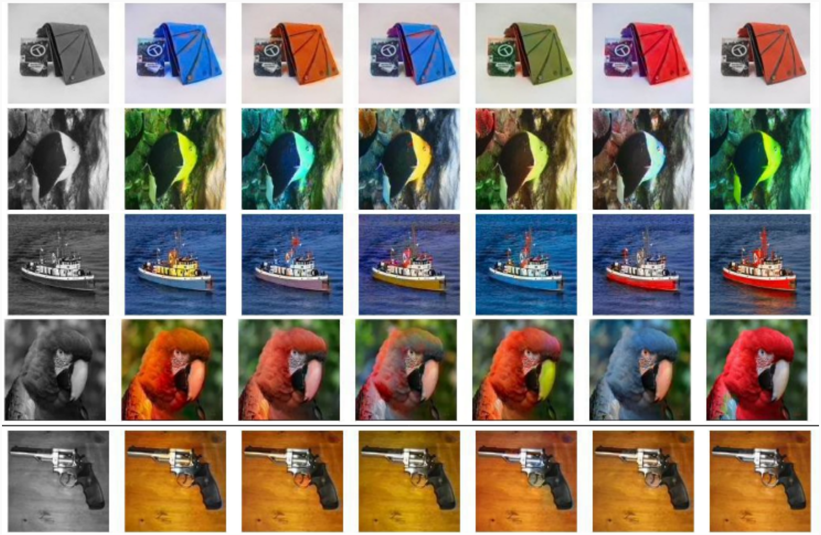


Model: $p(X^{\text{color}}|X^{\text{gray}})$

- Handles diversity and pixel correlations
- Clean objective, no ad-hoc heuristics

Parallel work: **PixColor: Pixel Recursive Colorization by Sergio Guadarrama et al**

Qualitative Results



Thank you for your attention!

Face generation demo:

`https://github.com/kolesman/FaceGeneration`

Image colorization code:

`https://github.com/ameroyer/PIC`

Apply for PhD at IST Austria: https://phd.pages.ist.ac.at/

Deadline: 8 January (very soon!)