Conditional Image Generation with PixelCNN Decoders

Aaron van den Oord[1] Nal Kalchbrenner[1] Oriol Vinyals[1] Lasse Espeholt[1] Alex Graves[1] Koray Kavukcuoglu[1]

[1]Google DeepMind

NIPS, 2016
Presenter: Beilun Wang

Outline

1. Introduction
   - Motivation
   - Previous Solutions
   - Contributions
2. Proposed Methods
   - PixelCNN
   - Conditional PixelCNN
3. Summary

Outline

1. Introduction
   - Motivation
   - Previous Solutions
   - Contributions
2. Proposed Methods
   - PixelCNN
   - Conditional PixelCNN
3. Summary

Motivation

Motivation:

- Conditional image generator with image density model.
- This generator can also be conditioned on class labels, descriptions and a single human face.
- Fast and parallel training.

Problem Setting:

Problem Setting:

- Input: Image with missing pixels, OR image class labels, OR a vector in the embedded space, OR a single human face
- Target: joint distribution consisting of conditional distribution with CNN.
- Output: Image
- PixelCNN (Pixel RNN):

$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, \ldots, x_{i-1}) \tag{1}$$

- Conditional Version:

$$p(\mathbf{x}|\mathbf{h}) = \prod_{i=1}^{n^2} p(x_i | x_1, \ldots, x_{i-1}, \mathbf{h}) \tag{2}$$

- B conditioned on (R,G); G conditioned on R.

Outline

1. Introduction
   - Motivation
   - Previous Solutions
   - Contributions

2. Proposed Methods
   - PixelCNN
   - Conditional PixelCNN

3. Summary

Previous Solutions

- PixelRNN

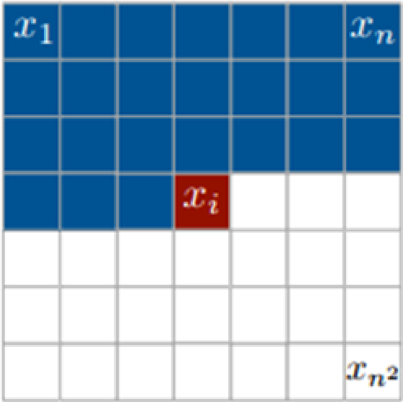


Figure: PixelRNN

Outline

1. Introduction
   - Motivation
   - Previous Solutions
   - Contributions
2. Proposed Methods
   - PixelCNN
   - Conditional PixelCNN
3. Summary

Contributions

- A fast and parallel trainable deep neural nets model for conditional image generator (?)
- Gated convolutional layers
- Conditional Gated convolutional layers

Outline

1. Introduction
   - Motivation
   - Previous Solutions
   - Contributions
2. Proposed Methods
   - PixelCNN
   - Conditional PixelCNN
3. Summary

Pixel CNN

- Input: $N \times N \times 3$
- Output: $N \times N \times 3 \times 256$

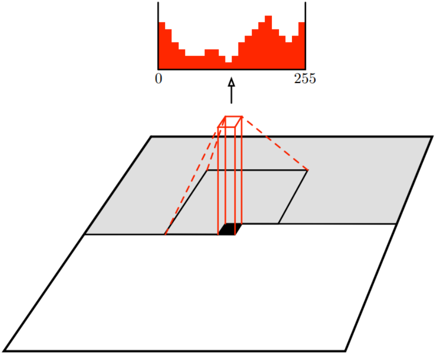
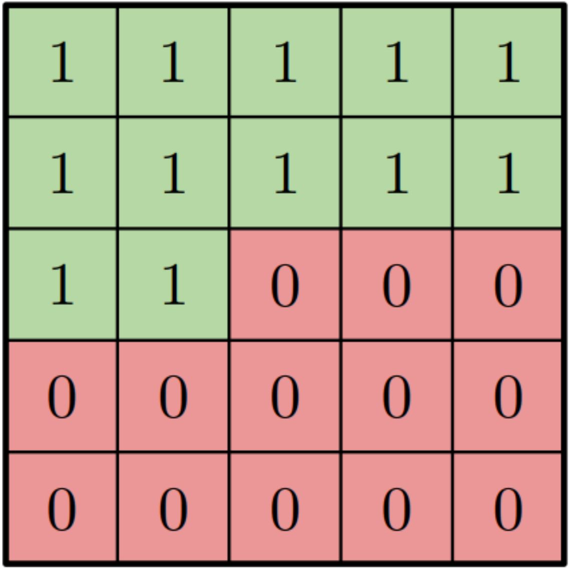


Figure: PixelCNN

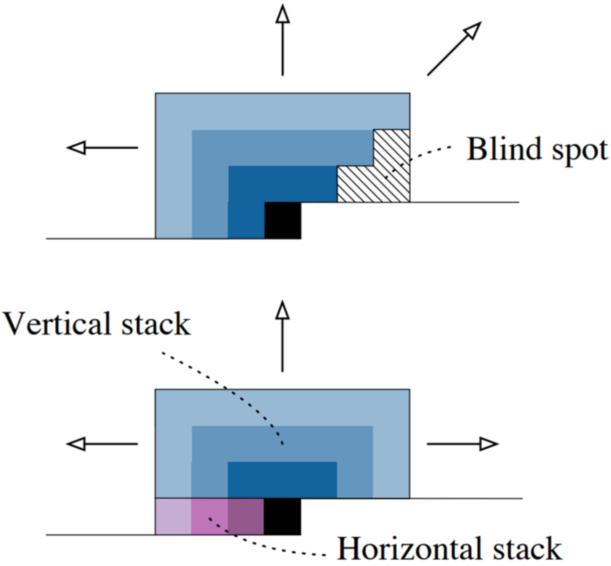
Only above and left Pixels are considered.

Masked Filter



1	1	1	1	1
1	1	1	1	1
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0

Blind spot



Gated Convolutional Layers

- The gates in LSTM may help it to model more complex interactions.
- This is also studied by paper like Highway networks, grid LSTM, and Neural GPUs.

$$\mathbf{y} = \tanh(W_{k,f} * \mathbf{x}) \odot \sigma(W_{k,g} * \mathbf{x}) \tag{3}$$

Gated Convolutional Layers – figure

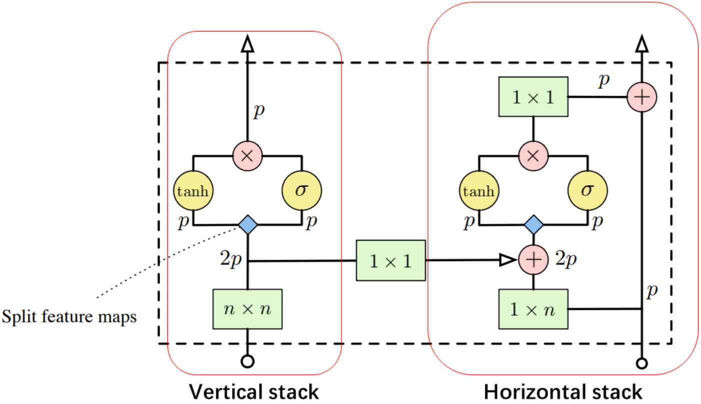


Figure: Gated Convolutional Layers

Outline

1. Introduction
   - Motivation
   - Previous Solutions
   - Contributions
2. Proposed Methods
   - PixelCNN
   - Conditional PixelCNN
3. Summary

Conditional PixelCNN

- replace $p(\mathbf{x}|\mathbf{h})$ from $p(\mathbf{x})$.

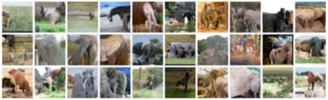
$$\mathbf{y} = \tanh(W_{k,f} * \mathbf{x} + V_{k,f}^T \mathbf{h}) \odot \sigma(W_{k,g} * \mathbf{x} + V_{k,g}^T \mathbf{h}) \tag{4}$$

Conditional PixelCNN–Embedded

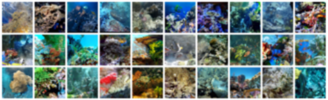
- Use a deconvolutional neural nets $m()$
- map $\mathbf{h}$ back to the image space as $\mathbf{s}$

$$\mathbf{y} = \tanh(W_{k,f} * \mathbf{x} + V_{k,f}^{T}\mathbf{s}) \odot \sigma(W_{k,g} * \mathbf{x} + V_{k,g}^{T}\mathbf{s}) \tag{5}$$

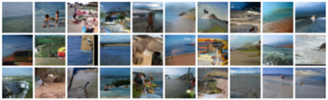
Experiment Results–image generation based on labels



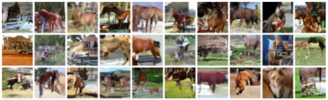
African elephant



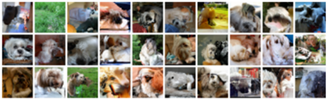
Coral Reef



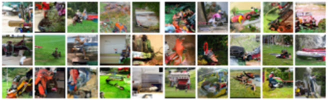
Sandbar



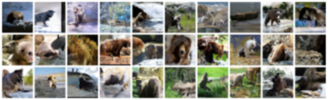
Sorrel horse



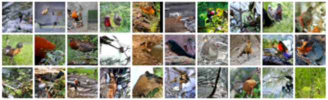
Lhasa Apso (dog)



Lawn mower

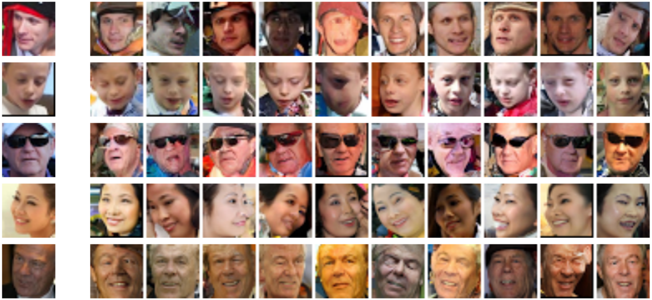


Brown bear



Robin (bird)

Experiment Results–image generation based on a single human face



Summary

- This paper improves the PixelCNN by the gated activation unit
- This paper extends the PixelCNN to a conditional version