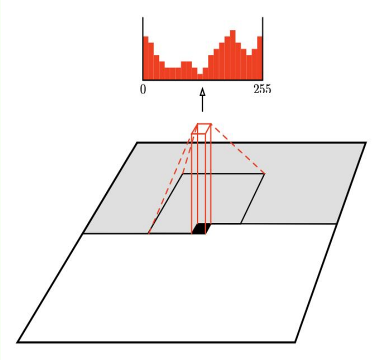
Conditional Image Generation with PixelCNN Decoders

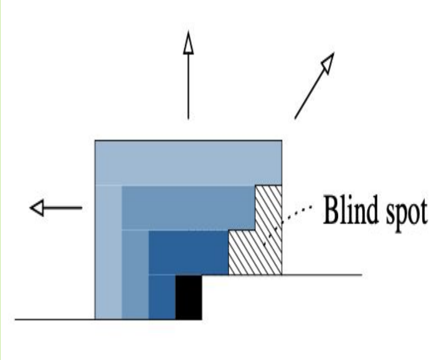
Mahshid Alinoori

Table of Contents

PixelCNN



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



PixelCNN models the joint distribution of pixels as:
$p(x) = \prod p(x_i|x_1, ..., x_{i-1})$.

Masked convolutions are used to remove the dependency upon future pixels.

Maske convolution results in ignoring some pixels of the input image so called **blind spot.**

Why PixelCNN Variants?

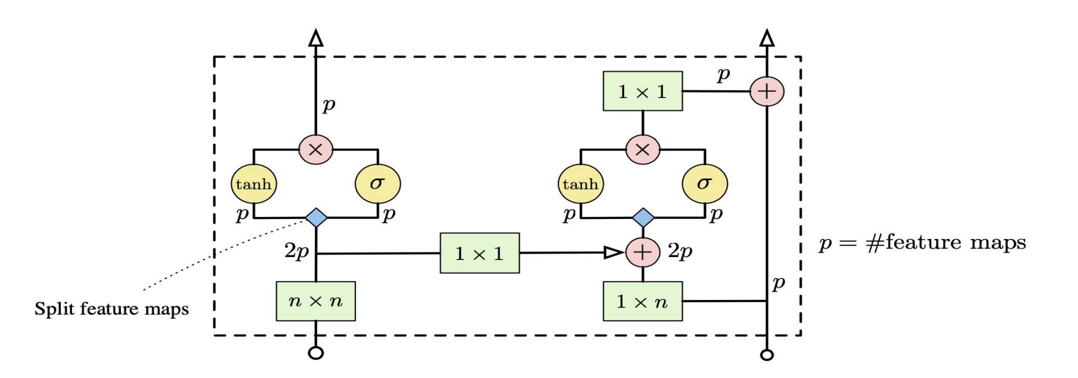
PixelCNN is **faster** than PixelRNN in training but **not as good in performance**

LSTMs have access to **entire neighborhood** of previous pixels and they can model more **complex interactions** using the gates..

Blind spot imposes miscalculation of the probability distribution

An updated version of PixelCNN is required to resolve the drawbacks

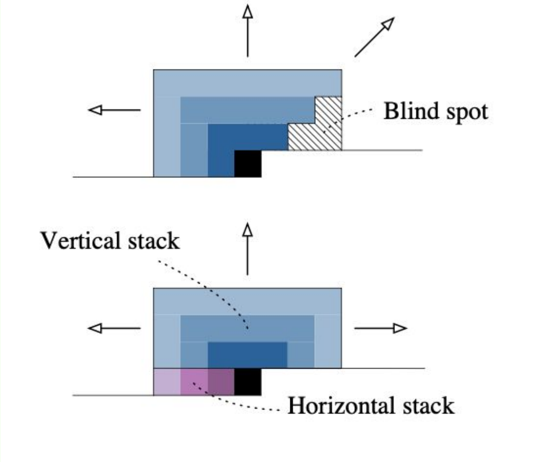
Gated PixelCNN



Relu activation is replaced by gated activation units:

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

Gated PixelCNN



Blind spots are addressed using two convolutional network stacks:

1. horizental stack: watching current row
2. vertical stack: watching all rows above

Gated PixelCNN Results

Gated PixelCNN outperforms PixelCNN by 0.11 bits/dim on CIFAR-10

Gated PixelCNN outperforms PixelRNN on ImageNet with half the training time required by PixelRNN

Conditional PixelCNN

- Conditional generation is used in **improving images** conditioned on noisy and incomplete data, **class-conditional modeling**, and generating variety of images with **similar features** to the input
- Conditional PixelCNN models the distribution p(x|h) as: **p(x|h) =∏ p(xi|x1, ..., xi−1, h)**
- Class conditioning activation is: $\mathbf{y=\tanh(W_{k,f}*x+V_{k,f}{}^{T}\,h)\odot\sigma(W_{k,g}*x+V_{k,g}{}^{T}\,h)}$
- location-based conditioning activation is: $\mathbf{y = \tanh(W_{k,f}*x+V_{k,f}*s)\odot\sigma(W_{k,g}*x+V_{k,g}*s)}$ where **s = m(h)** and Vk,g is 1x1 convolution

Conditional PixelCNN Results

Class-conditional medelling of ImageNet using one-hot encoding of the class results in **higher visual quality**

Grey whale

Generating new portraits of same person using embeddings from a CNN as the condition h

Generating images conditioned on **linear interpolation** between embeddings of pairs of images

PixelCNN AutoEncoder

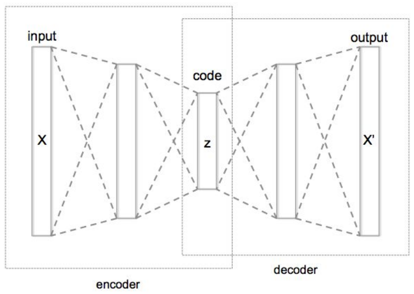
Decoders in auto-encoders try to model the distribution **p(x|h)** and Conditional PixelCNN seems good at this job

The decoder is replaced by a PixelCNN. It takes care of low-level statistics and the encoder can focus on **high-level information**

PixelcNN auto-encoder generates images of **more variety and higher quality** compared to conventional auto-encoders

$m = 10$

$m = 100$



Conclusion

This paper focused on 3 items with respect to PixelCNN:

- Introducing Gated PixelCNN as an improvement to the original PixelCNN
- Introducing Conditional PixelCNN to address conditional image generation
- Application of Conditional PixelCNN as image decoders in building the auto-encoders

Thanks!