Deep Generative Models Supplementary

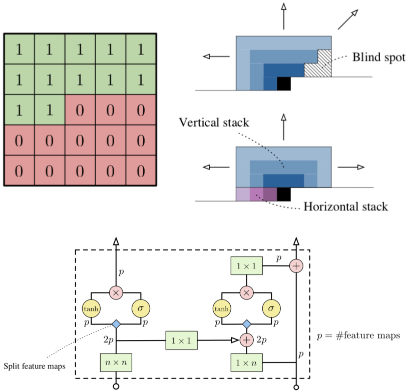
Roman Isachenko

Moscow Institute of Physics and Technology

2020

=======

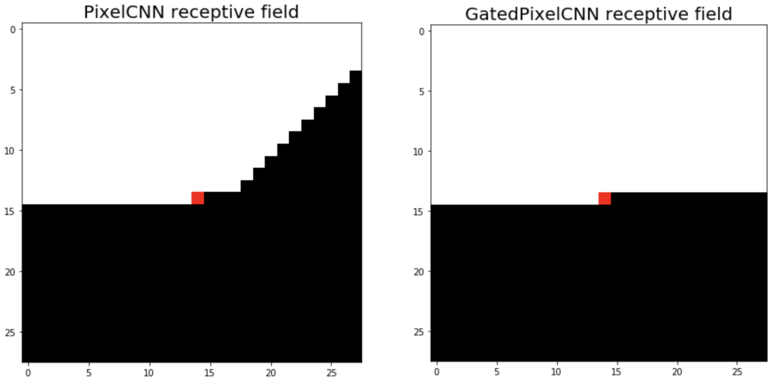
GatedPixelCNN (2016)



Van den Oord A. et al. Conditional image generation with pixelcnn decoders

https://arxiv.org/pdf/1606.05328.pdf

GatedPixelCNN (2016)



Van den Oord A. et al. Conditional image generation with pixelcnn decoders
https://arxiv.org/pdf/1606.05328.pdf

Extensions

- **PixelCNN++**: *Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications* https://arxiv.org/pdf/1712.09763.pdf (mixture of logistics instead of softmax);
- **PixelSNAIL**: *An Improved Autoregressive Generative Model* https://arxiv.org/pdf/1712.09763.pdf (self-attention to learn optimal autoregression ordering).

=======

Summary

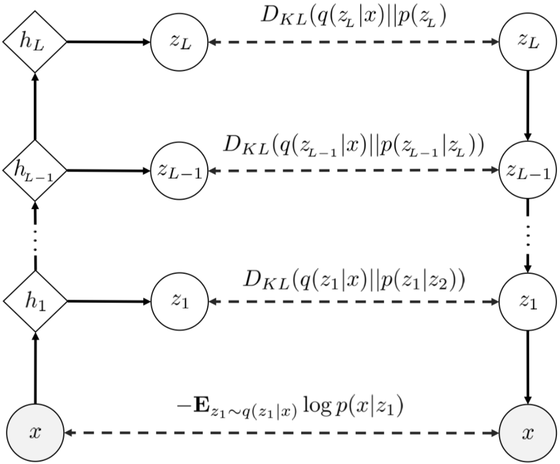
- Sampling from autoregressive models are trivial, but sequential
  - sample $x_0 \sim p(x_0)$;
  - sample $x_1 \sim p(x_1|x_0)$;
  - $\dots$.
- Estimating probability:

$$p(\mathbf{x}) = \prod_{i=1}^{m} p(x_i|\mathbf{x}_{1:i-1}).$$

- Work on both continuous and discrete data.
- There is no natural way to do unsupervised learning.

PixelVAE, 2016

Hierarchical VAE



https://arxiv.org/pdf/1611.05013.pdf

PixelVAE, 2016

Hierarchical decomposition

$$p(\mathbf{z}_1, \dots, \mathbf{z}_L) = p(\mathbf{z}_L) p(\mathbf{z}_{L-1}|\mathbf{z}_L) \dots p(\mathbf{z}_1, \mathbf{z}_2);$$

$$q(\mathbf{z}_1, \dots, \mathbf{z}_L|\mathbf{x}) = q(\mathbf{z}_1|\mathbf{x}) \dots q(\mathbf{z}_L|\mathbf{x}).$$

ELBO

$$\begin{aligned}
\mathcal{L}(q, \boldsymbol{\theta}) &= \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}_1, \boldsymbol{\theta}) - KL(q(\mathbf{z}_1, \dots, \mathbf{z}_L|\mathbf{x})||p(\mathbf{z}_1, \dots, \mathbf{z}_L)) \\
&= \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}_1, \boldsymbol{\theta}) - \int \prod_{j=1}^{L} q(\mathbf{z}_j|\mathbf{x}) \sum_{i=1}^{L} \log \frac{q(\mathbf{z}_i|\mathbf{x})}{p(\mathbf{z}_i|\mathbf{z}_{i+1})} d\mathbf{z}_1 \dots d\mathbf{z}_L \\
&= \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}_1, \boldsymbol{\theta}) - \sum_{i=1}^{L} \int \prod_{j=1}^{L} q(\mathbf{z}_j|\mathbf{x}) \log \frac{q(\mathbf{z}_i|\mathbf{x})}{p(\mathbf{z}_i|\mathbf{z}_{i+1})} d\mathbf{z}_1 \dots d\mathbf{z}_L \\
&= \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}_1, \boldsymbol{\theta}) - \sum_{i=1}^{L} \int q(\mathbf{z}_{i+1}|\mathbf{x}) q(\mathbf{z}_i|\mathbf{x}) \log \frac{q(\mathbf{z}_i|\mathbf{x})}{p(\mathbf{z}_i|\mathbf{z}_{i+1})} d\mathbf{z}_i d\mathbf{z}_{i+1} \\
&= \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}_1, \boldsymbol{\theta}) - \sum_{i=1}^{L} \mathbb{E}_{q(\mathbf{z}_{i+1}|\mathbf{x})} \left[KL(q(\mathbf{z}_i|\mathbf{x})||p(\mathbf{z}_i|\mathbf{z}_{i+1}))\right]
\end{aligned}$$

https://arxiv.org/pdf/1611.05013.pdf

References

- **MADE**: *Masked Autoencoder for Distribution Estimation*
https://arxiv.org/pdf/1502.03509.pdf
Summary: Create masked autoencoder that models autoregression (autoregression allows to make the distribution properly normalized). Sampling is performed iteratively (to generate MNIST image 784 forward passes are needed). Discrete data.
- **PixelRNN + PixelCNN**: *Pixel recurrent neural networks*
https://arxiv.org/abs/1601.06759
Summary: 2 models are proposed: PixelRNN, PixelCNN. The models are autoregression and sampling is sequential. For RNN two types of LSTM blocks are used: Row LSTM and DiagonalBiLSTM. CNN uses Masked convolutions. RNN outperforms, but is slower.
- **GatedPixelCNN**: *Conditional Image Generation with PixelCNN Decoders*
https://arxiv.org/pdf/1606.05328.pdf
Summary: Improvements for PixelCNN: gated units (like in lstm), horizontal+vertical stacks (remove blind spots). The result is now similar to PixelRNN.
- **WaveNet**: *a Generative Model for Raw Audio*
https://arxiv.org/pdf/1609.03499.pdf
Summary: Model for autoregressive audio generation, inspired by PixelCNN. Use causal convolutions for the right conditioning, and dilated atrous convolution to extend receptive field.
- **PixelCNN++**: *Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications*
https://arxiv.org/pdf/1701.05517.pdf
Summary: Improved version of PixelCNN. Models mixture of logistic mixture distribution instead of softmax. Architectural modifications: skip connections, up/down sampling, dropout. Experiment with dequantization: discretization works better.
- **PixelSNAIL**: *An Improved Autoregressive Generative Model*
https://arxiv.org/pdf/1712.09763.pdf
Summary: Autoregressive model. Uses masked causal convolutions. Adjust self-attention to PixelCNN.