Pixel Recurrent Neural Networks

Полина Кириченко

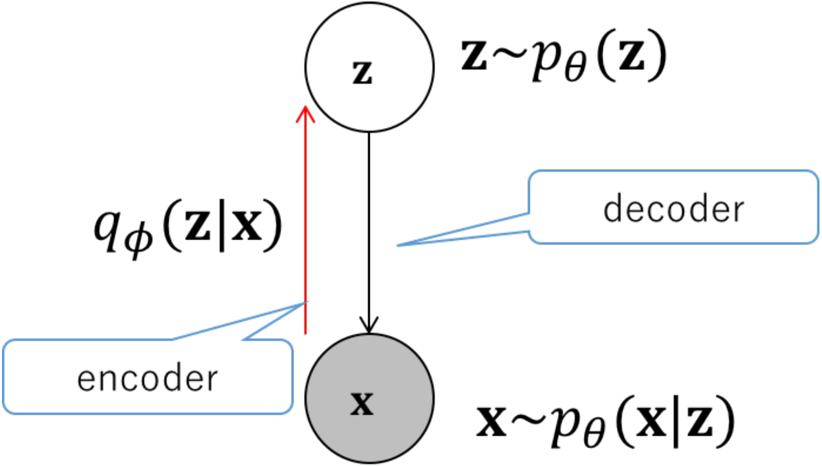
ФКН ВШЭ, 2017

Генерация естественных изображений

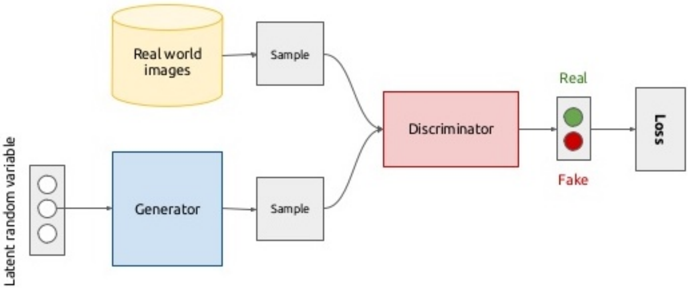
Цель: обучить генеративную модель естественных изображений

- Модели со скрытыми переменными (например, вариационный автокодировщик)
- Adversarial (GAN)
- Autoregressive (Pixel RNN) моделирование распредедения $p(X) = \prod_{i=1}^{n^2} p(x_i | x_{i-1}, \ldots, x_1)$

Вариационный автокодировщик

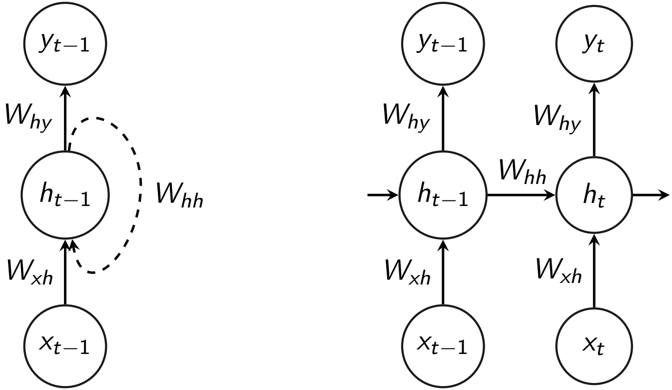


Generative Adversarial Networks

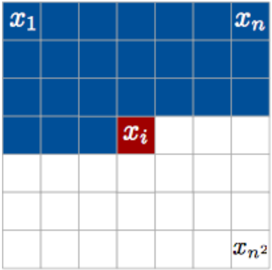


Recurrent Neural Network

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b)$$



Pixel RNN



$$p(X) = \prod_{i=1}^{n^2} p(x_i | x_{i-1}, \dots x_1)$$

$$p(x_{i,R}|\mathbf{x}_{<i})p(x_{i,G}|\mathbf{x}_{<i}, x_{i,R})p(x_{i,B}|\mathbf{x}_{<i}, x_{i,R}, x_{i,G})$$

- Похоже на language modelling
- Дискретное преставление пискелей (softmax на последнем слое)

Softmax sampling

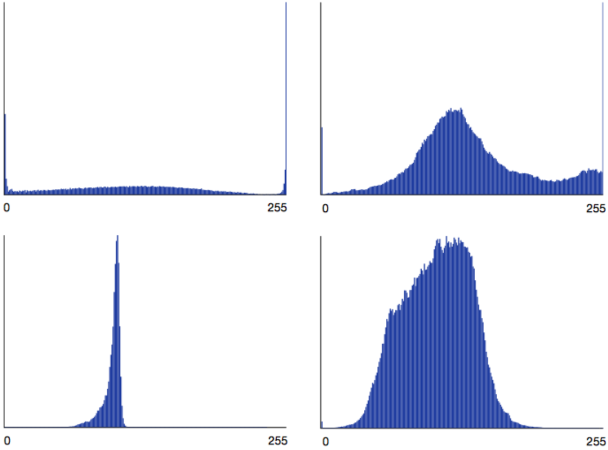


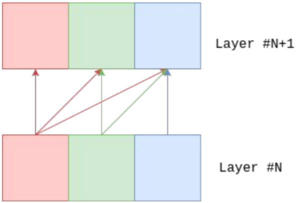
Рис.: Примеры активаций softmax слоя в модели: в модели нет априорных предположений о форме распределения.

Модели

- Pixel CNN
  - Fully convolutional
- Pixel RNN
  - Row LSTM
  - Diagonal Bi-LSTM

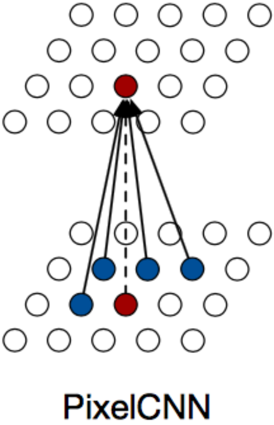
Masked Convolution

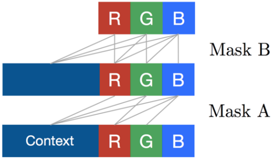
1	1	1	1	1
1	1	1	1	1
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0



Mask type 'B'

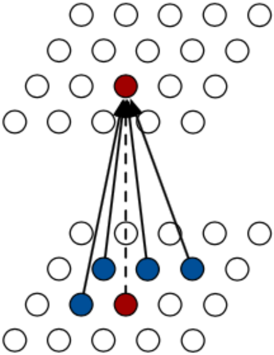
Pixel CNN





No pooling layers!

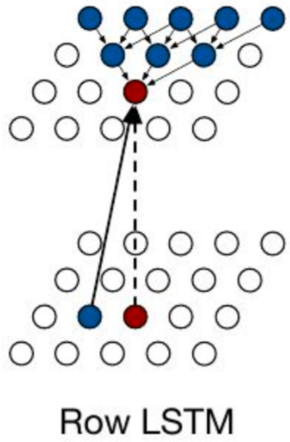
Pixel CNN



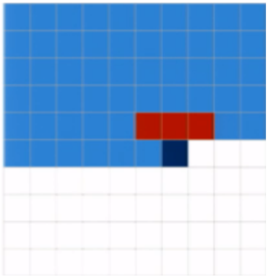
PixelCNN

- Предсказание и backbrop через все пиксели одновременно
- Распараллеливание из-за сверток, самая быстрая архитектура из трех
- Фиксированный receptive field :(

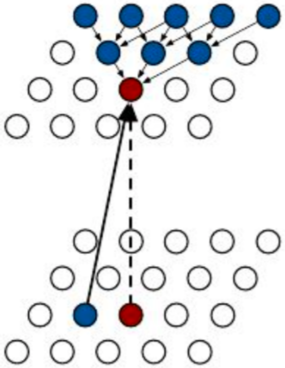
Row LSTM



- Recurrent connections
- State is a whole vector
- Convolutional state-to-state mapping



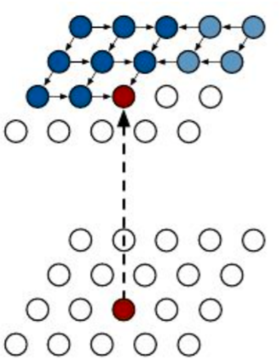
Row LSTM



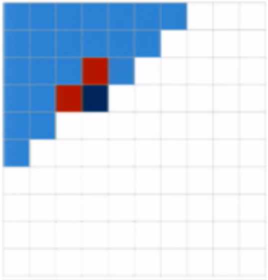
Row LSTM

$$[\mathbf{o}_i, \mathbf{f}_i, \mathbf{i}_i, \mathbf{g}_i] = \sigma(\mathbf{K}^{ss} \circledast \mathbf{h}_{i-1} + \mathbf{K}^{is} \circledast \mathbf{x}_i)$$
$$\mathbf{c}_i = \mathbf{f}_i \odot \mathbf{c}_{i-1} + \mathbf{i}_i \odot \mathbf{g}_i$$
$$\mathbf{h}_i = \mathbf{o}_i \odot \tanh(\mathbf{c}_i)$$

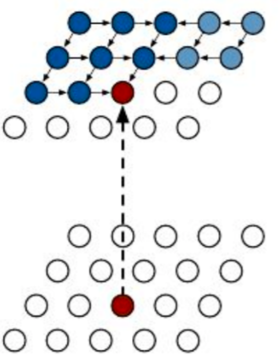
Diagonal Bi-LSTM

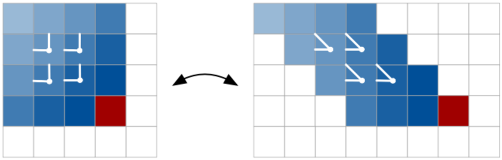


- State progresses diagonally
- Captures entire available context
- Bidirectional: 2 LSTMS (from top left and top right)



Diagonal Bi-LSTM





$$[\mathbf{o}_i, \mathbf{f}_i, \mathbf{i}_i, \mathbf{g}_i] = \sigma(\mathbf{K}^{ss} \circledast \mathbf{h}_{i-1} + \mathbf{K}^{is} \circledast \mathbf{x}_i)$$

$$\mathbf{c}_i = \mathbf{f}_i \odot \mathbf{c}_{i-1} + \mathbf{i}_i \odot \mathbf{g}_i$$

$$\mathbf{h}_i = \mathbf{o}_i \odot \tanh(\mathbf{c}_i)$$

Архитекутры

PixelCNN	Row LSTM	Diagonal BiLSTM
7×7 conv mask A		
Multiple residual blocks: (see fig 5)		
Conv 3×3 mask B	Row LSTM i-s: 3×1 mask B s-s: 3×1 no mask	Diagonal BiLSTM i-s: 1×1 mask B s-s: 1×2 no mask
ReLU followed by 1×1 conv, mask B (2 layers)		
256-way Softmax for each RGB color (Natural images) or Sigmoid (MNIST)		

Residual blocks

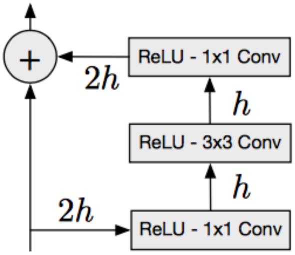


Рис.: Residual block for CNN

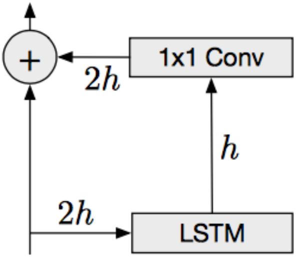


Рис.: Residual block for RNN

Residuals blocks: experiments

# layers:	1	2	3	6	9	12
NLL:	3.30	3.20	3.17	3.09	3.08	3.06

Рис.: Negative log likelihood evaluated on the CIFAR-10 validation set

	No skip	Skip
No residual:	3.22	3.09
Residual:	3.07	3.06

Рис.: Negative log likelihood evaluated on the CIFAR-10 validation set, 12 layers

CIFAR-10

Model	**NLL Test**
DBM 2hl [1]:	≈ 84.62
DBN 2hl [2]:	≈ 84.55
NADE [3]:	88.33
EoNADE 2hl (128 orderings) [3]:	85.10
EoNADE-5 2hl (128 orderings) [4]:	84.68
DLGM [5]:	≈ 86.60
DLGM 8 leapfrog steps [6]:	≈ 85.51
DARN 1hl [7]:	≈ 84.13
MADE 2hl (32 masks) [8]:	86.64
DRAW [9]:	$\lesssim 80.97$
PixelCNN:	81.30
Row LSTM:	80.54
Diagonal BiLSTM (1 layer, $h = 32$):	**80.75**
Diagonal BiLSTM (7 layers, $h = 16$):	**79.20**

Occluded images

Результаты

- Autoregressive models for image generation
- Use 2D-LSTM (Pixel RNN) and convolutions (Pixel CNN) to model conditional distribution
- Softmax for pixel prediction
- Masked convolutions enforce ordering and model color dependecies
- State-of-the art (log-likelihood) for Binary MNIST, CIFAR-10 and realistic generated samples

Reference

Статья: https://arxiv.org/abs/1601.06759