Generative Image Modeling Using Spatial LSTMs

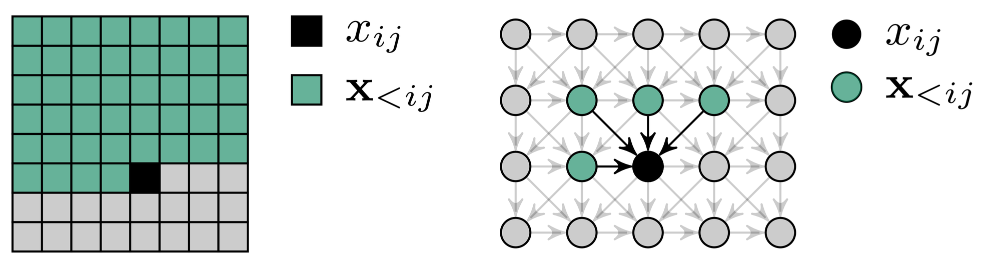
Lucas Theis[1,2], Matthias Bethge[2,3]

[1] Graduate School of Neural Information Processing, University of Tübingen
[2] Werner Reichardt Centre for Integrative Neuroscience, Tübingen
[3] Max Planck Institute for Biological Cybernetics, Tü

Introduction

We introduce a tractable image model based on a combination of multi-dimensional recurrent neural networks [1] and a specific mixture of experts [2]. Quantitative comparisons show that the model outperforms the state of the art in natural image density estimation.

Directed graphical modeling



The directed modeling approach turns the density estimation problem into a supervised problem of learning $p(x_{ij} \mid \mathbf{x}_{<ij})$.

$$p(\mathbf{x}) = \prod_{i,j} p(x_{ij} \mid \mathbf{x}_{<ij})$$

This approach has been shown to work very well for natural images [e.g., 2, 3, 4].

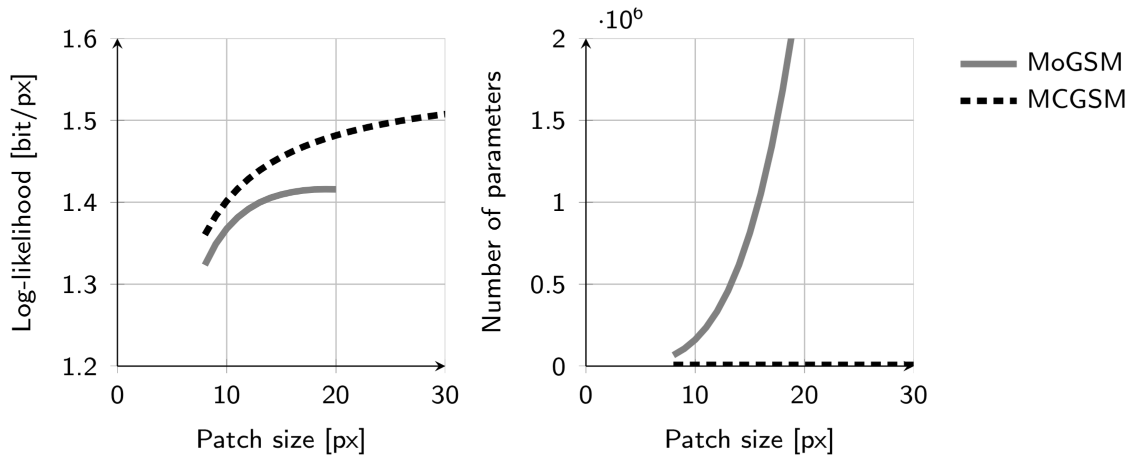
Factorized mixtures of conditional GSMs

As a basis for our model, we use a factorized form of the MCGSM [2]:

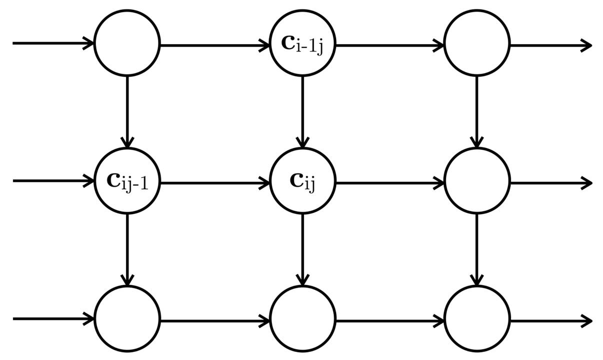
$$p(x_{ij} \mid \mathbf{x}_{<ij}) = \sum_{c,s} \underbrace{p(c, s \mid \mathbf{x}_{<ij})}_{\text{gate}} \underbrace{p(x_{ij} \mid \mathbf{x}_{<ij}, c, s)}_{\text{expert}}$$

$$p(c, s \mid \mathbf{x}_{<ij}) \propto \exp\left(\eta_{cs} - \tfrac{1}{2} e^{\alpha_{cs}} \textstyle\sum_n \beta_{cn}^2 (\mathbf{b}_n^\top \mathbf{x}_{<ij})^2\right.$$
$$p(x_{ij} \mid \mathbf{x}_{<ij}, c, s) = \mathcal{N}(x_{ij}; \mathbf{a}_c^\top \mathbf{x}_{<ij}, e^{-\alpha_{cs}})$$

The model generalizes mixtures of GSMs (MoGSM) but scales much better to large images:



Spatial LSTMs



We use multi-dimensional recurrent neural networks [1] to transform the neighborhoods $\mathbf{x}_{<ij}$ into hidden state vectors $\mathbf{h}_{ij}$:

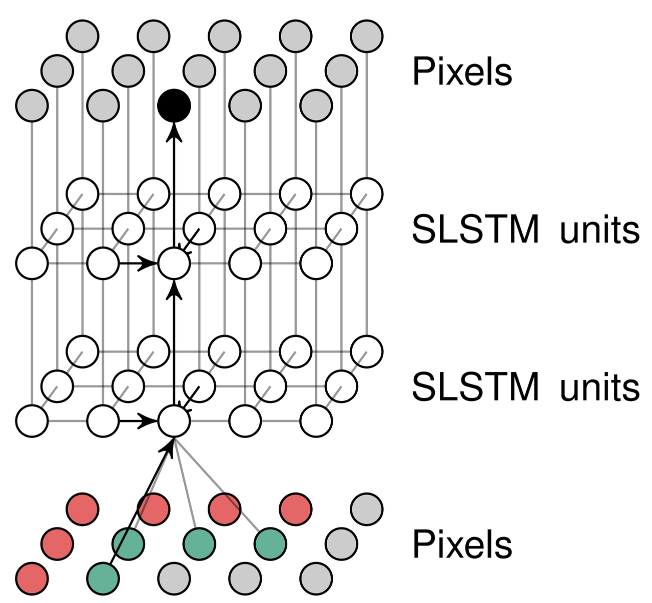
$$\mathbf{c}_{ij} = \mathbf{g}_{ij} \odot \mathbf{i}_{ij} + \mathbf{c}_{i,j-1} \odot \mathbf{f}_{ij}^c + \mathbf{c}_{i-1,j} \odot \mathbf{f}_{ij}^r$$
$$\mathbf{h}_{ij} = \tanh(\mathbf{c}_{ij} \odot \mathbf{o}_{ij})$$

where

$$\begin{pmatrix} \mathbf{g}_{ij} \\ \mathbf{o}_{ij} \\ \mathbf{i}_{ij} \\ \mathbf{f}_{ij}^r \\ \mathbf{f}_{ij}^c \end{pmatrix} = \begin{pmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \\ \sigma \end{pmatrix} T_{\mathbf{A},\mathbf{b}} \begin{pmatrix} \mathbf{x}_{<ij} \\ \mathbf{h}_{i,j-1} \\ \mathbf{h}_{i-1,j} \end{pmatrix}$$

Recurrent image density estimator

We combine the MCGSM with spatial LSTMs to form the recurrent image density estimator (RIDE), $p(x_{ij} \mid \mathbf{x}_{<ij}) = p(x_{ij} \mid \mathbf{h}_{ij})$.

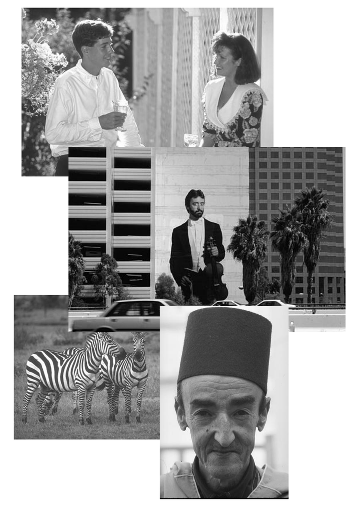


Ensembles

To further improve performance, we form ensembles over transformed models/images (e.g. rotation, flipping):

$$q(\mathbf{x}) = \frac{1}{K} \sum_k p(\mathbf{T}_k \mathbf{x}) |\det \mathbf{T}_k|$$

Density estimation (natural images)



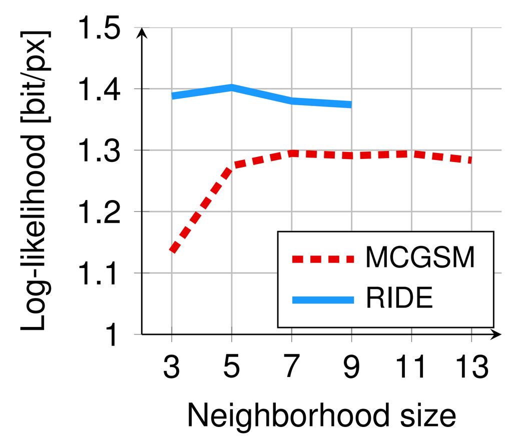
BSDS 300

Model	63 dim. [nat]	64 dim. [bit/px]	∞ dim. [bit/px]
RNADE	152.1	3.346	-
RNADE, 1 hl	143.2	3.146	-
RNADE, 6 hl	155.2	3.416	-
EoRNADE, 6 layers	157.0	3.457	-
GMM, 200 comp.	153.7	3.360	-
STM, 200 comp.	155.3	3.418	-
Deep GMM, 3 layers	156.2	3.439	-
MCGSM, 16 comp.	155.1	3.413	3.688
MCGSM, 32 comp.	155.8	3.430	3.706
MCGSM, 64 comp.	156.2	3.439	3.716
MCGSM, 128 comp.	156.4	3.443	3.717
EoMCGSM, 128 comp.	**158.1**	**3.481**	3.748
RIDE, 1 layer	150.7	3.293	3.802
RIDE, 2 layers	152.1	3.346	3.869
EoRIDE, 2 layers	154.5	3.400	**3.899**

Density estimation (dead leafs)

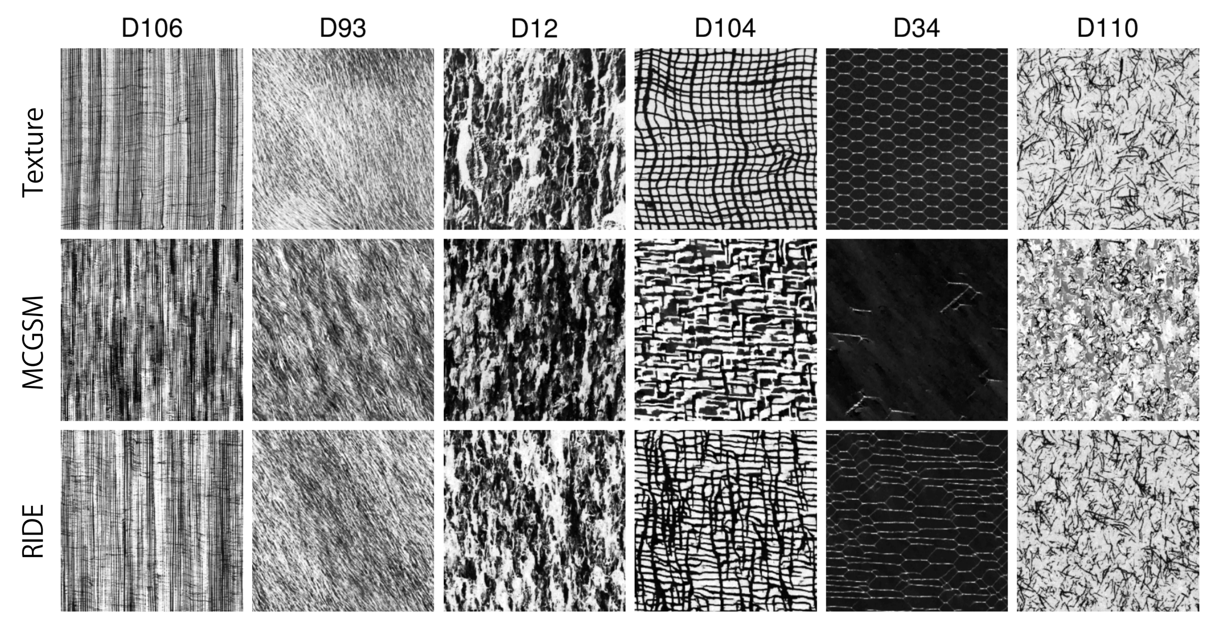
The right-hand plot shows the performance of an MCGSM and RIDE as a function of neighborhood size. The saturation of the MCGSM deonstrates that the better performance of RIDE is not just due to the indirect access to more pixels, but that the nonlinear transformation matters.

Model	[bit/px]
MCGSM, 12 comp.	1.244
MCGSM, 32 comp.	1.294
Diffusion	1.489
RIDE, 64 hid., 1 layer	1.402
RIDE, 64 hid., 1 layer, ext.	1.416
RIDE, 64 hid., 2 layers	1.438
RIDE, 64 hid., 3 layers	1.454
RIDE, 128 hid., 3 layers	1.489
EoRIDE, 128 hid., 3 layers	**1.501**



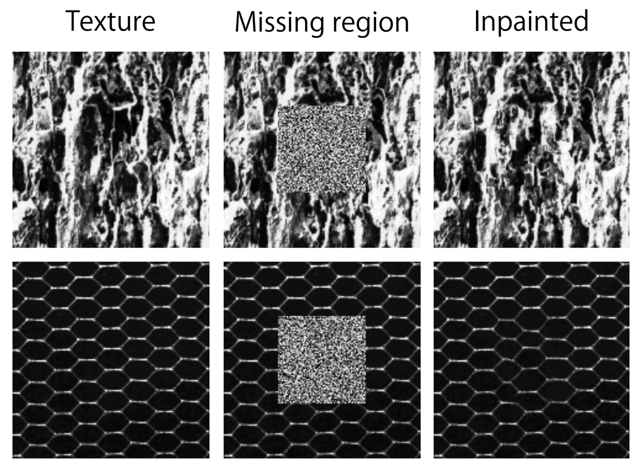
Texture synthesis

We trained the factorized MCGSM and RIDE on individual Brodatz textures [5]. Textures not seen during training and samples are shown below:



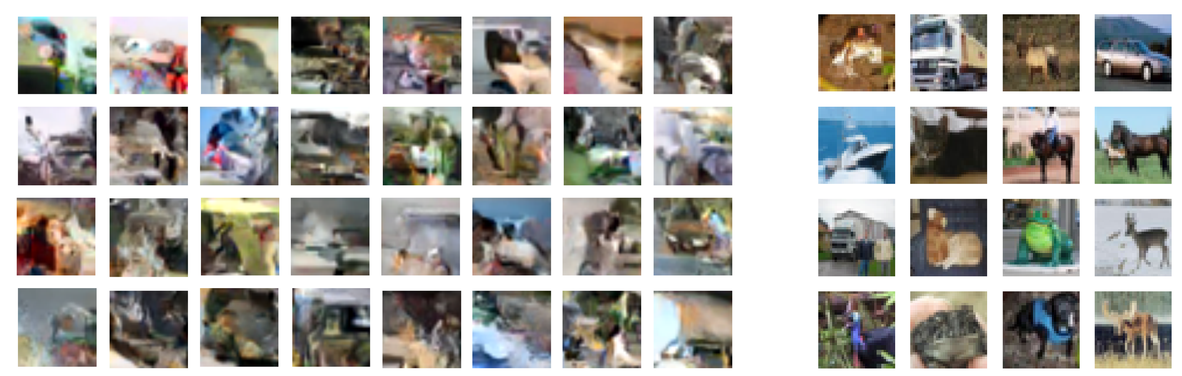
Texture inpainting

We used Metropolis within Gibbs sampling to inpaint 71 x 71 pixel regions in textures:



CIFAR-10 samples

Samples of RIDE trained on 32 x 32 pixel images:



RIDE, 3 layers   CIFAR-10

Discussion

- Deep and recurrent neural networks can improve image density estimation
- Although our model is computationally tractable, it is still slow to train (recurrent structure not a good fit for GPU)
- In future work we therefore want to explore alternative deep extensions of the MCGSM

Code

Python/caffe implementation of RIDE :

http://github.com/lucastheis/ride/

References

[1] A. Graves and J. Schmidhuber, NIPS, 2009
[2] L. Theis, R. Hosseini, and M. Bethge, PLoS ONE, 2012
[3] R. Hosseini, F. Sinz, and M. Bethge, Vision Research, 2010
[4] B. Uria, I. Murray, and H. Larochelle, NIPS, 2013
[5] P. Brodatz, 1966, http://www.ux.uis.no/~tranden/brodatz.html