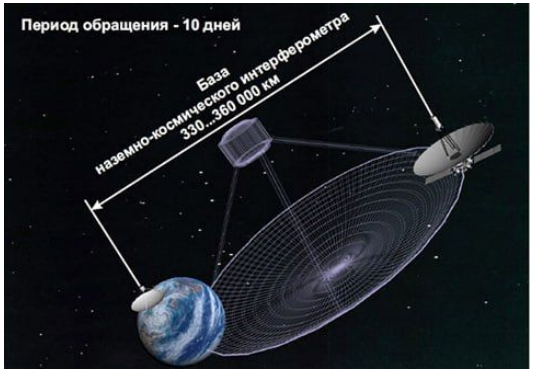
Повышение разрешения потоковых нейросетевых **моделей для обработки** данных РСДБ **наблюдений** проекта **Радиоастрон**

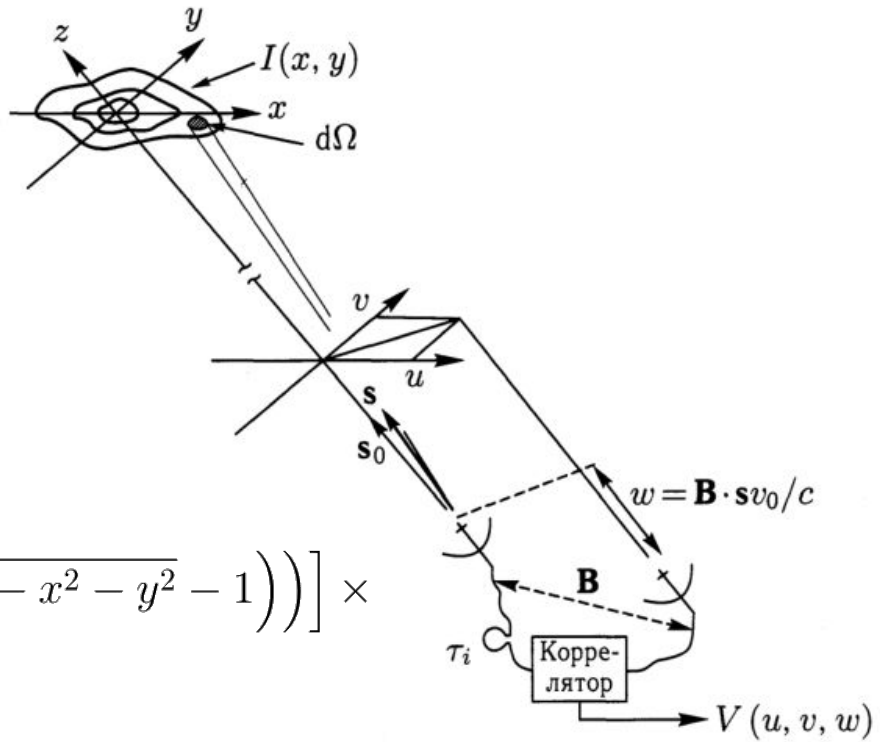
Кац Лев

Лаборатория математических методов обработки АКЦ ФИАН

Задача восстановления изображения



$$y = f(x)$$



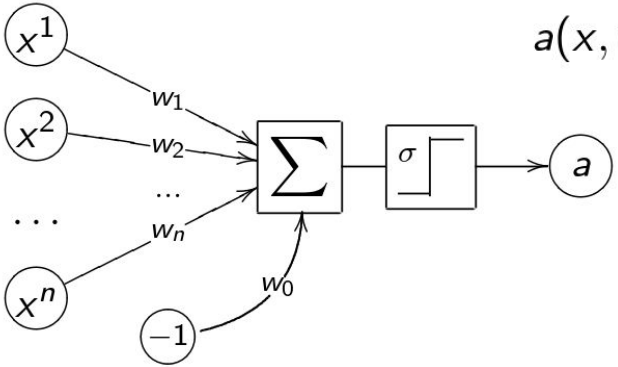
$$V(u,v,w) = \iint\limits_{x^2+y^2<1} A(x,y)I(x,y)\times$$

$$\times\, exp\left[-2\pi i\left(ux + vy + w\left(\sqrt{1-x^2-y^2}-1\right)\right)\right]\times$$

$$\times\, \frac{dxdy}{\sqrt{1-x^2-y^2}}$$

Инструменты и методы радиоастрономии Т. Л. Уилсон, К. Рольфс, С. Хюттемейстер. - Москва : Физматлит.

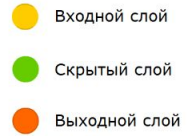
Модель нейрона.

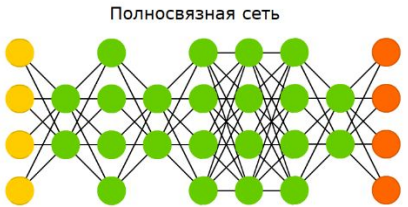


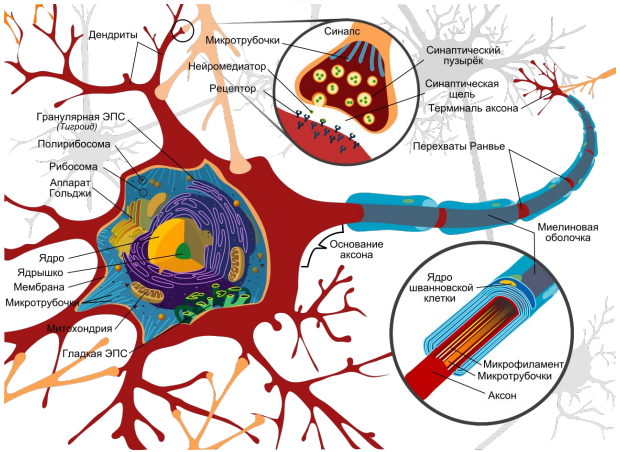
$$a(x,w) = \sigma\big(\langle w,x\rangle\big) = \sigma\left(\sum_{j=1}^{n} w_j f_j(x) - w_0\right),$$

МакКаллок-Питтс, 1943

σ(z) — **функция активации.** Она обязательно **нелинейная;**
w_j — **весовые коэффициенты;**
w_0 — **порог активации нейрона.**







Теорема Цыбенко (1989): Нейронная сеть с **одним скрытым слоем** может равномерно **аппроксимировать любую** непрерывную **функцию** на отрезке с заданной точностью

Постановка задачи. Апостериорное распределение

$$\theta^\star = \arg\min_\theta D_{\mathrm{KL}}(q_\theta(x)\|p(x|y))$$

$$= \arg\min_\theta \int q_\theta(x)[\log q_\theta(x) - \log p(x|y)]dx$$

$$\dots$$

$$x \sim q_\theta(x) \quad \Leftrightarrow \quad x = G_\theta(z), z \sim \mathcal{N}(0,1)$$

$$\dots$$

$$= \arg\min_\theta \mathbb{E}_{z\sim\mathcal{N}(0,1)}\{-\log p(y|G_\theta(z)) - \log p(G_\theta(z))$$

$$+ \log \pi(z) - \log \left|\det \frac{dG_\theta(z)}{dz}\right|\Big\}.$$

Расстояние **Ку́льбака — Ле́йблера**

$$D_{\mathrm{KL}}(P \parallel Q) = \int_X p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x,$$

, где **P**, **Q** — плотности распределения

$$D_{\mathrm{KL}}(P \parallel Q) \geq 0,$$

$$D_{KL}(P \parallel Q) = 0 \iff P = Q$$

$$D_{\mathrm{KL}}(P \parallel Q) \neq D_{\mathrm{KL}}(Q \parallel P).$$

arxiv: 2010.14462

Регуляризации как плотность распределения

$$\theta^\star = \arg\min_\theta \mathbb{E}_{z\sim\mathcal{N}(0,1)}\{-\log p(y|G_\theta(z)) - \log p(G_\theta(z)) + \log \pi(z) - \log\left|\det \frac{dG_\theta(z)}{dz}\right|\}$$

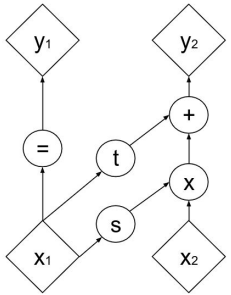
$$\theta^\star = \arg\min_\theta \sum_{k=1}^{N}\{\mathcal{L}(y, f(G_\theta(z_k))) + \lambda\mathcal{R}(G_\theta(z_k)) - \log\left|\det \frac{dG_\theta(z_k)}{dz_k}\right|\},$$

$$-\log p\left(G_\theta(z)\right) \longrightarrow \lambda\,\mathfrak{R}\left(G_\theta(z)\right)$$

$$\begin{aligned}\mathfrak{R}(x) &= \lambda_1\,\text{l1\_loss}(x) + \lambda_2\,\text{loss\_TSV}(x) + \lambda_3\,\text{loss\_center}(x) + \\ &+ \lambda_4\,\text{loss\_flux}(x) + \lambda_5\,\text{loss\_ce}(x)\end{aligned}$$

arxiv: 2010.14462

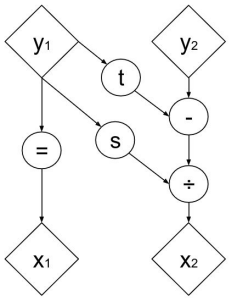
Affine Coupling layer и Real NVP



forward

$$y_{1:d} = x_{1:d}$$
$$y_{d+1:D} = x_{d+1:D} \odot \exp\left(s(x_{1:d})\right) + t(x_{1:d}),$$

$$\frac{\partial y}{\partial x^T} = \begin{bmatrix} \mathbb{I}_d & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \operatorname{diag}\left(\exp\left[s\left(x_{1:d}\right)\right]\right) \end{bmatrix},$$



inverse

$$\det \frac{\partial y}{\partial x^T} = \exp\left[\sum_j s\left(x_{1:d}\right)_j\right].$$

arxiv: 1605.08803

Увеличение размеров генерируемых изображений

Glow: Generative Flow with Invertible 1×1 Convolutions

Diederik P. Kingma*, **Prafulla Dhariwal***
OpenAI, San Francisco

Abstract

Flow-based generative models (Dinh et al., 2014) are conceptually a
tractability of the exact log-likelihood, tractability of exact latent-var
and parallelizability of both training and synthesis. In this paper we propose *Glow*,
a simple type of generative flow using an invertible 1 ×
method we demonstrate a significant improvement in lo
benchmarks. Perhaps most strikingly, we demonstrate
optimized towards the plain log-likelihood objective is ca
looking synthesis and manipulation of large images. Th
available at https://github.com/openai/glow.

1 Introduction

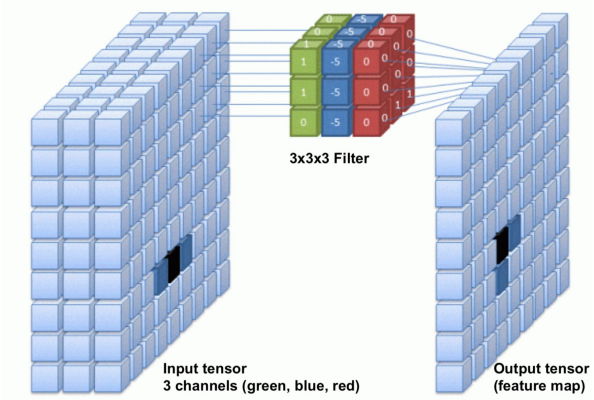
Two major unsolved problems in the field of machine learning ar
learn from few datapoints, like humans; and (2) generalization: robustness to changes of the task or
its context. AI systems, for example, often do not work at all when given inputs that are different
from their training distribution. A promise of *generative models*, a major branch of machine learning,

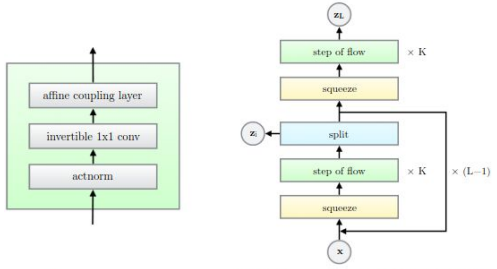
*Equal contribution.

$$\log\left|\det\left(\frac{d\,\mathtt{conv2D}(\mathbf{h};\mathbf{W})}{d\,\mathbf{h}}\right)\right| = h\cdot w\cdot\log\left|\det(\mathbf{W})\right|$$

$$\mathbf{W} = \mathbf{PL}(\mathbf{U} + \mathrm{diag}(\mathbf{s}))$$

$$\log\left|\det(\mathbf{W})\right| = \mathtt{sum}(\log|\mathbf{s}|)$$





(a) One step of our flow.

(b) Multi-scale architecture (Dinh et al., 2016).

arxiv: 1807.03039

Увеличения размеров генерируемых изображений

Wavelet Flow: Fast Training of High Resolution Normalizing Flows

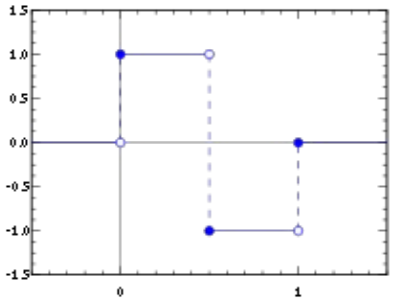
Jason J. Yu[1,3], **Konstantinos G. Derpanis**[2,4,5] **and Marcus A. Brubaker**[1,3,5]
[1]Department of Electrical Engineering and Computer Science, York University, Toronto
[2]Department of Computer Science, Ryerson University, Toronto
[3]Borealis AI, [4]Samsung AI Centre Toronto, [5]Vector Institute
jjyu@eecs.yorku.ca kosta@cs.ryerson.ca mab@eecs.yorku.ca

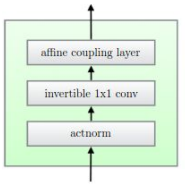
Abstract

Normalizing flows are a class of probabilistic generative models which allow for both fast density computation and efficient sampling and are effective at modelling complex distributions like images. A drawback among current methods is their significant training cost, sometimes requiring months of GPU training time to achieve state-of-the-art results. This paper introduces *Wavelet Flow*, a multi-scale, normalizing flow architecture based on wavelets. A Wavelet Flow has an explicit representation of signal scale that inherently includes models of lower resolution signals and conditional generation of higher resolution signals, *i.e.*, super resolution. A major advantage of Wavelet Flow is the ability to construct generative models for high resolution data (*e.g.*, 1024×1024 images) that are impractical with previous models. Furthermore, Wavelet Flow is competitive with previous normalizing flows in terms of bits per dimension on standard (low resolution) benchmarks while being up to $15\times$ faster to train.

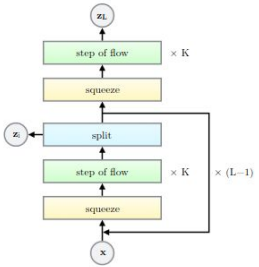
1 Introduction

Here we introduce *Wavelet Flow*, a multi-scale, conditional normalizing flow architecture based on

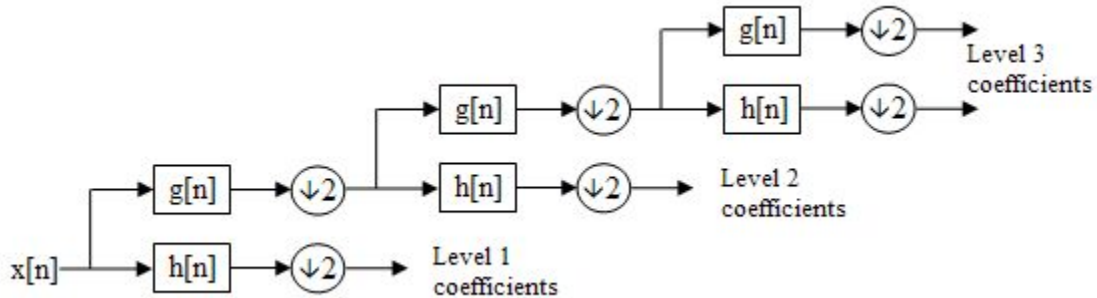




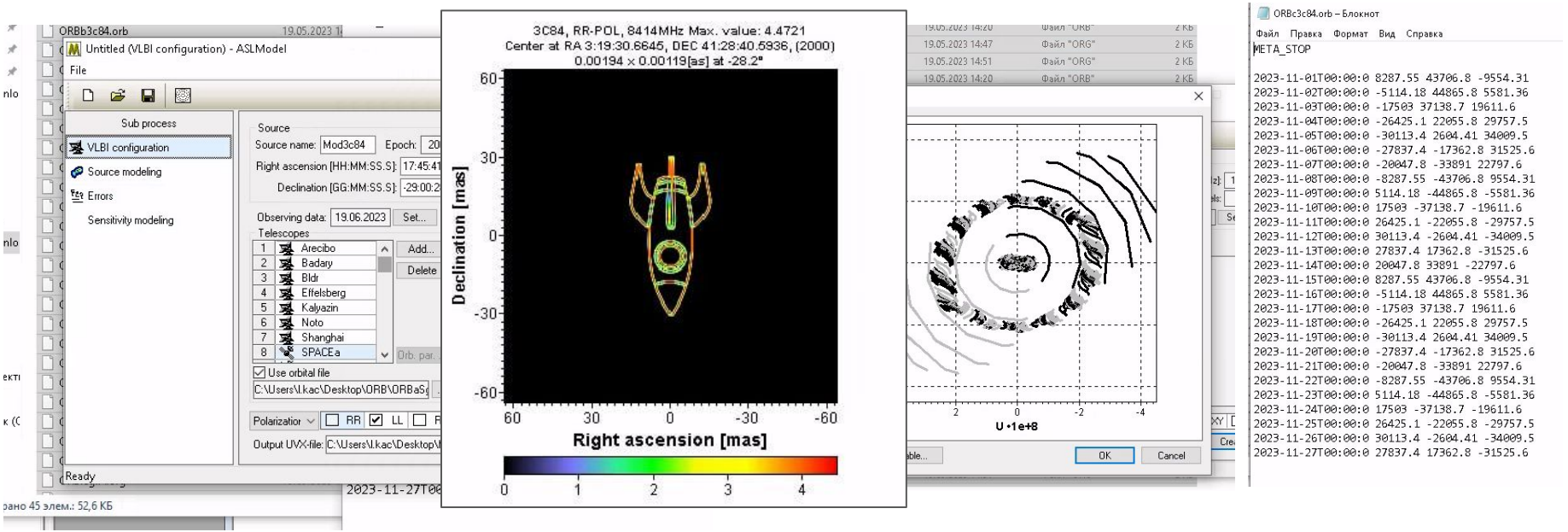
(a) One step of our flow.



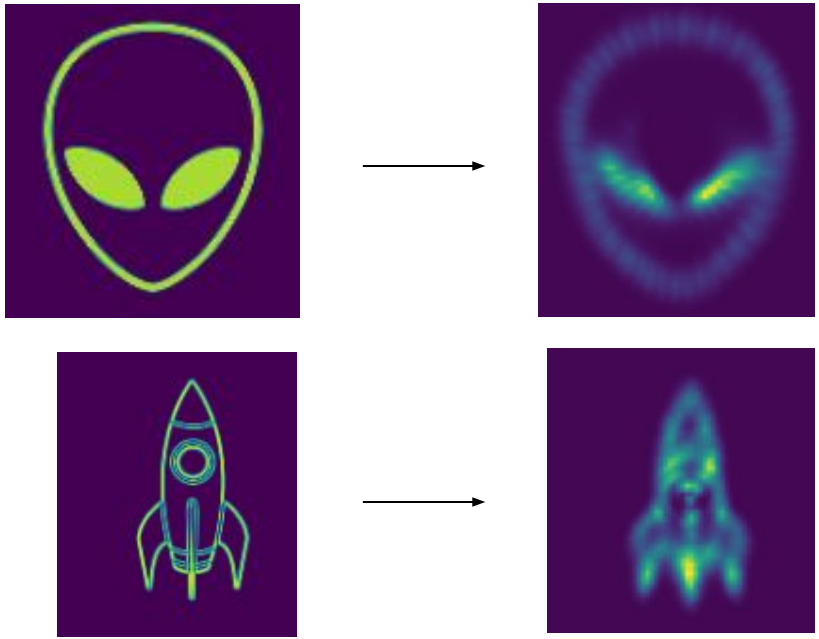
(b) Multi-scale architecture (Dinh et al., 2016).



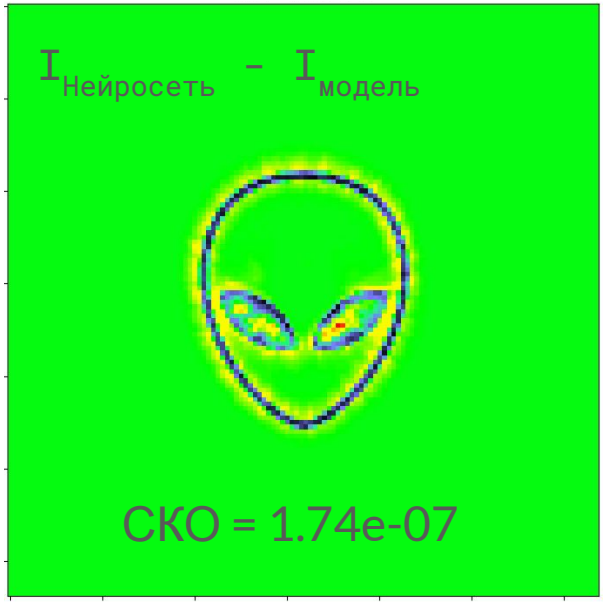
Создание модельных датасетов.

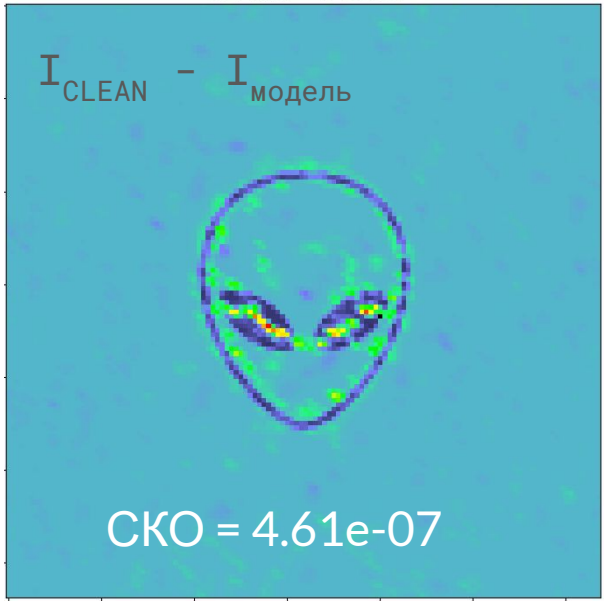


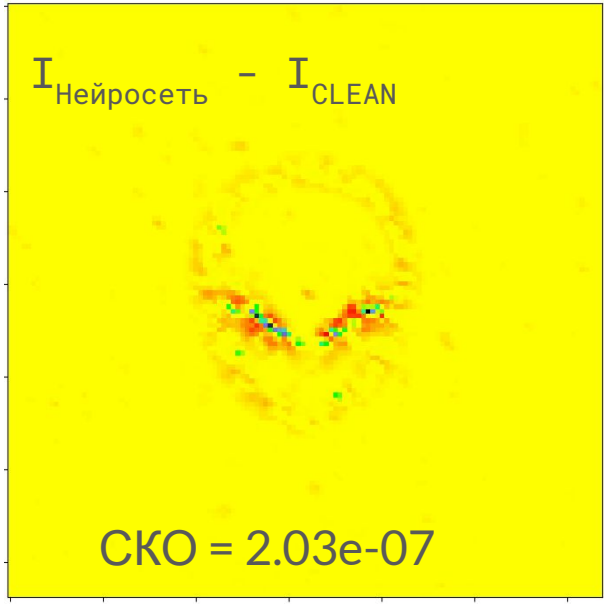
Восстановление модельных изображений



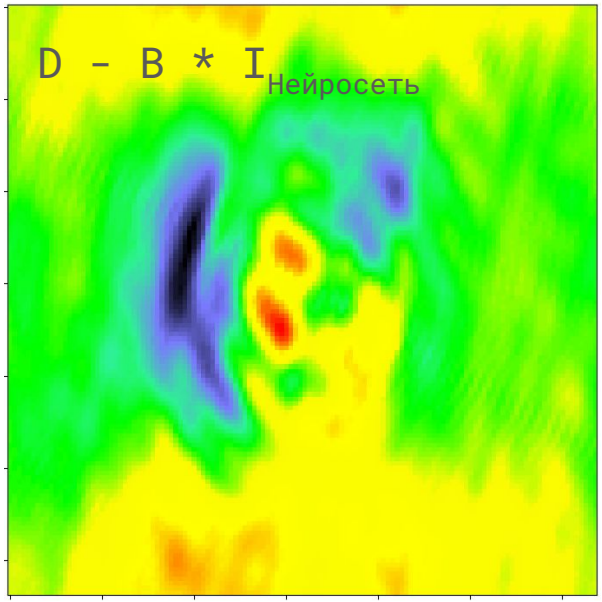
Сравнение с классическими методами

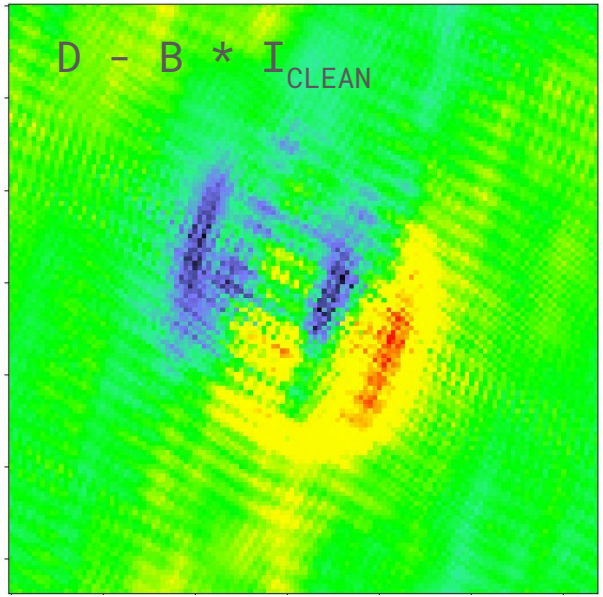


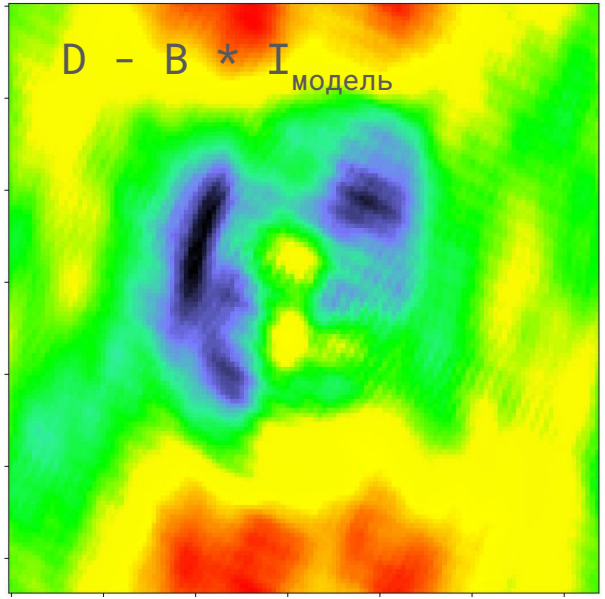




Сравнение с классическими методами

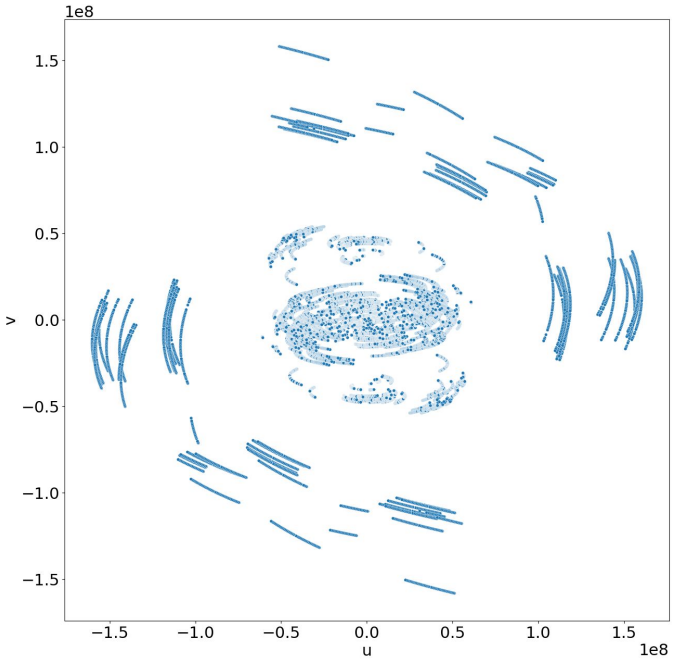




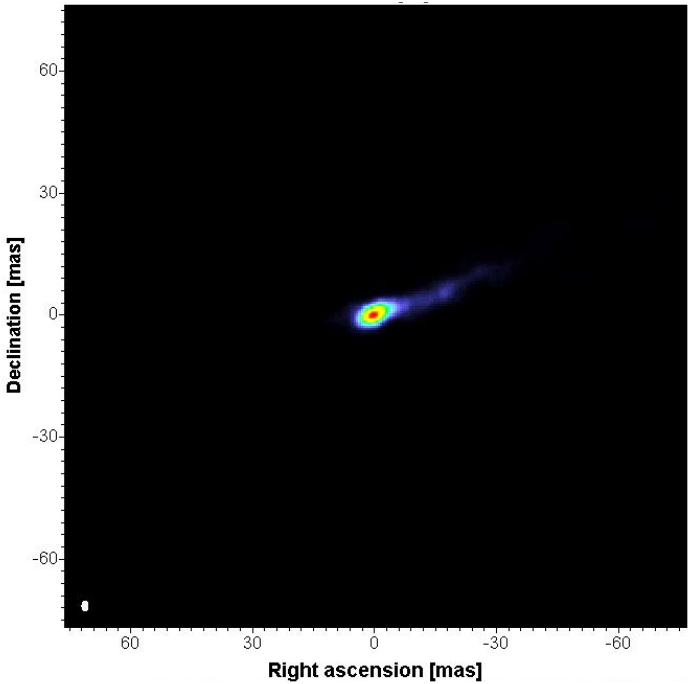


Данные наблюдений M87.

1228+126 LL-POL
1652.25 MHz
RA 12:30:49.4 (2000)
DEC 12:23:28.4 (2000)



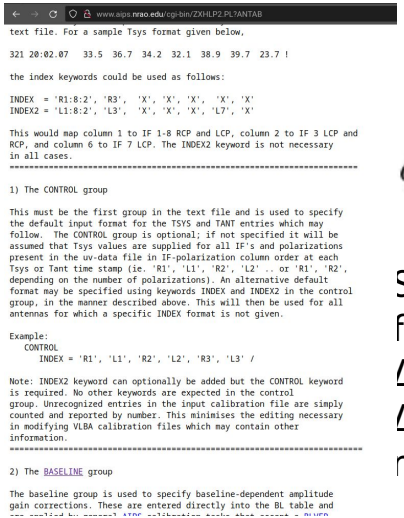
CLEAN



Данные наблюдений. Шумы

$$\tilde{V}_{ik}(t) = g_i(t) g_k^*(t) V_{ik} + \varepsilon_{ik}(t).$$

$\tilde{V}_{ik}$ — наблюдаемая видность
V_{ik} — истинная видность


www.aips.nrao.edu/cgi-bin/ZXHLP2.PL?ANTAB

text file. For a sample Tsys format given below,

321 20:02.07 33.5 36.7 34.2 32.1 38.9 39.7 23.7 !

the index keywords could be used as follows:

INDEX = 'R1:8:2', 'R3', 'X', 'X', 'X', 'X', 'X'
INDEX2 = 'L1:8:2', 'L3', 'X', 'X', 'X', 'L7', 'X'

This would map column 1 to IF 1-8 RCP and LCP, column 2 to IF 3 LCP and RCP, and column 6 to IF 7 LCP. The INDEX2 keyword is not necessary in all cases.

---

1) The CONTROL group

This must be the first group in the text file and is used to specify the default input format for the TSYS and TANT entries which may follow. The CONTROL group is optional; if not specified it will be assumed that Tsys values are supplied for all IF's and polarizations present in the uv-data file in IF-polarization column order at each Tsys or Tant time stamp (ie. 'R1', 'L1', 'R2', 'L2' .. or 'R1', 'R2', depending on the number of polarizations). An alternative default format may be specified using keywords INDEX and INDEX2 in the control group, in the manner described above. This will then be used for all antennas for which a specific INDEX format is not given.

Example:
CONTROL
INDEX = 'R1', 'L1', 'R2', 'L2', 'R3', 'L3' /

Note: INDEX2 keyword can optionally be added but the CONTROL keyword is required. No other keywords are expected in the control group. Unrecognized entries in the input calibration file are simply counted and reported by number. This minimises the editing necessary in modifying VLBA calibration files which may contain other information.

---

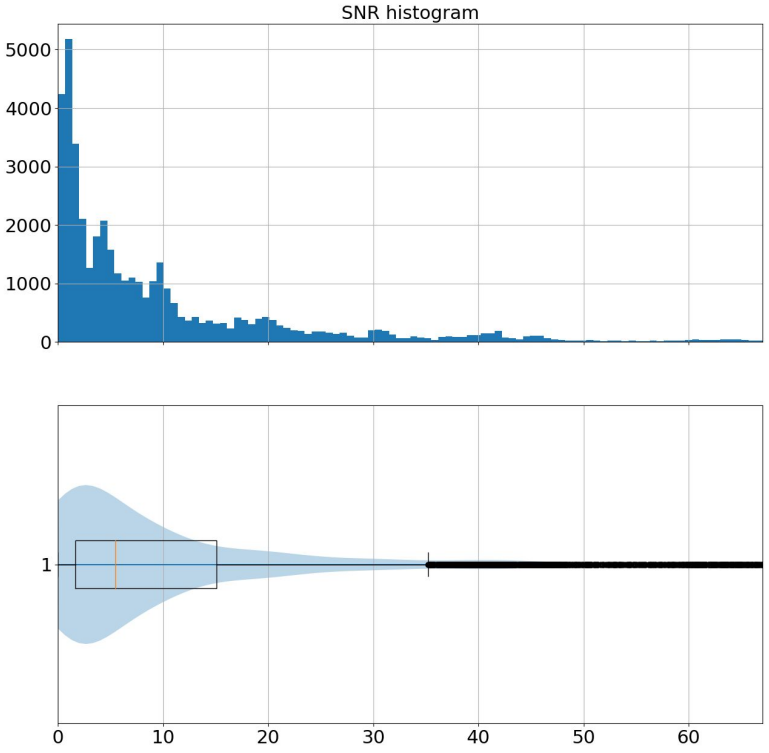
2) The BASELINE group

The baseline group is used to specify baseline-dependent amplitude gain corrections. These are entered directly into the BL table and

Тепловой шум:

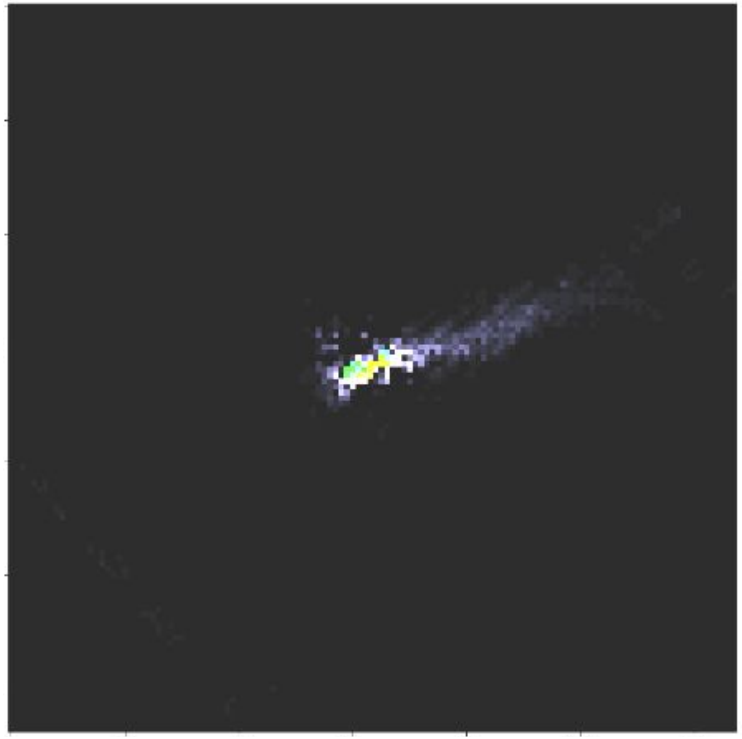
$$\sigma_{ij} = \frac{1}{\eta} \sqrt{\frac{\mathrm{SEFD}_i \times \mathrm{SEFD}_j}{2 \Delta\nu \, \Delta t}}$$

SEFD — system equivalent flux density;
Δν — ширина полосы;
Δt — интервал интегрирования;
η = 0.88.



arxiv: 1803.07088 + эта работа

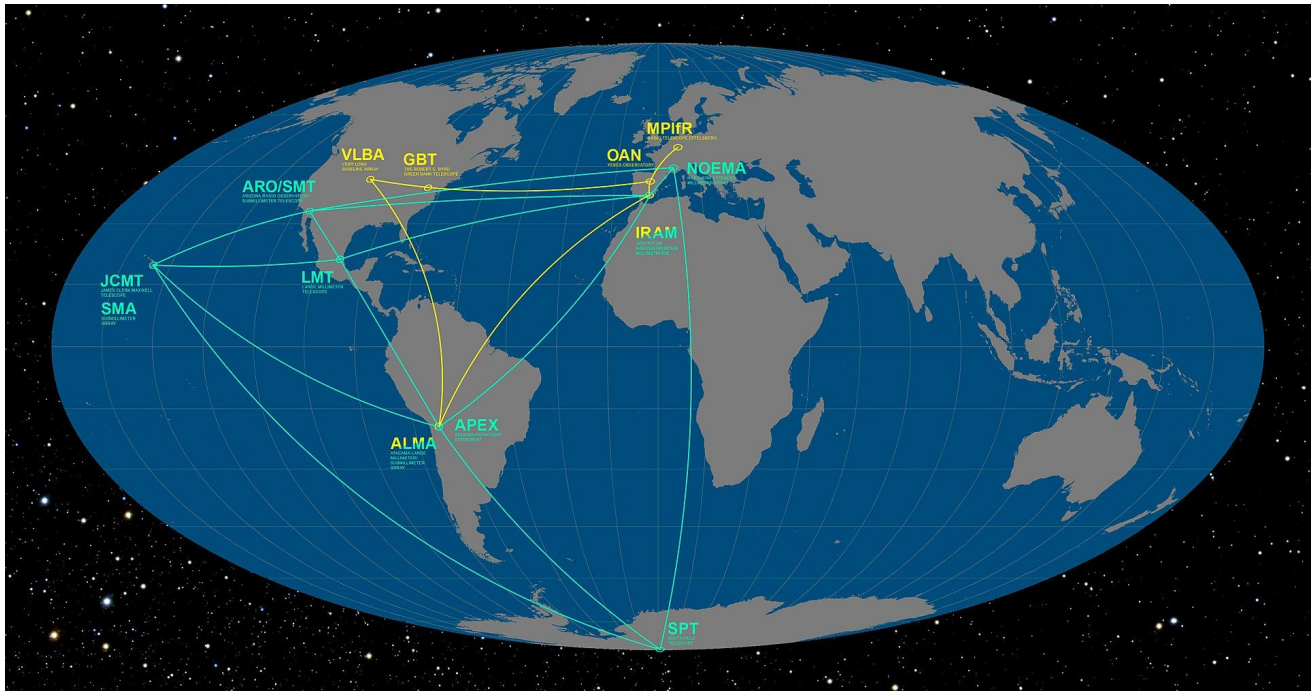
Данные наблюдений M87.



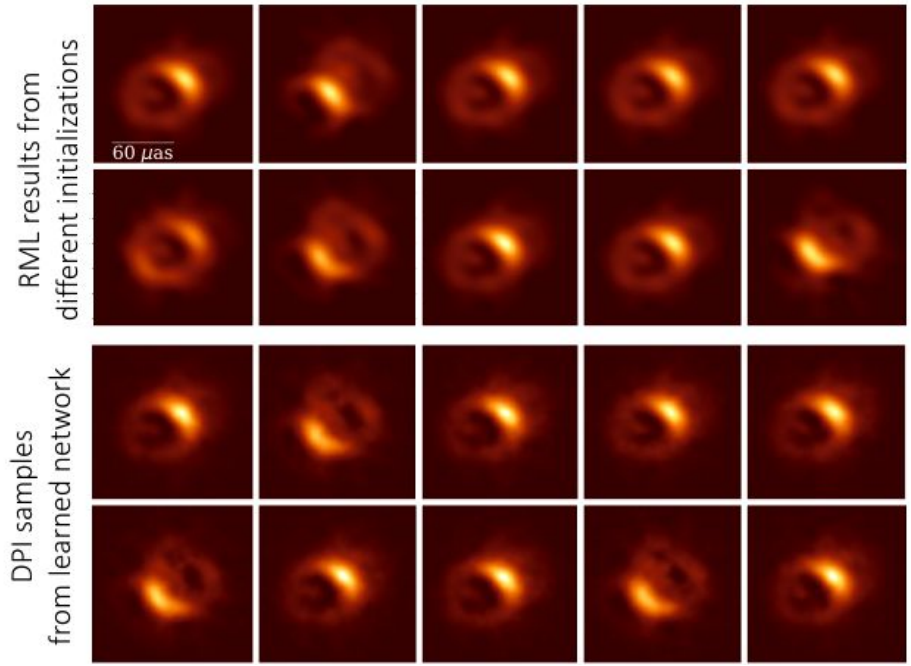
Планы

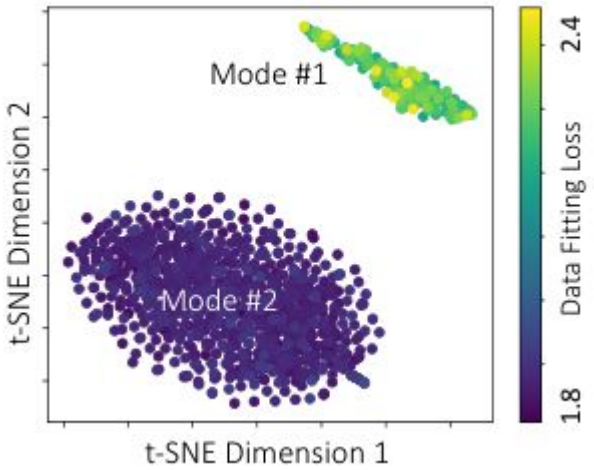
1. Дальнейшее **повышение разрешения** (**сейчас 256x256**), вероятно потребует распределенного обучения
2. Подробное **исследование распределения изображений**, а не отдельных реализаций
3. **Учет систематических** мультипликативных **погрешностей** из-за неточной **калибровки** (обычно для этого используют **соотношения замыканий**)

Спасибо за внимание!



Достоверность деталей изображения





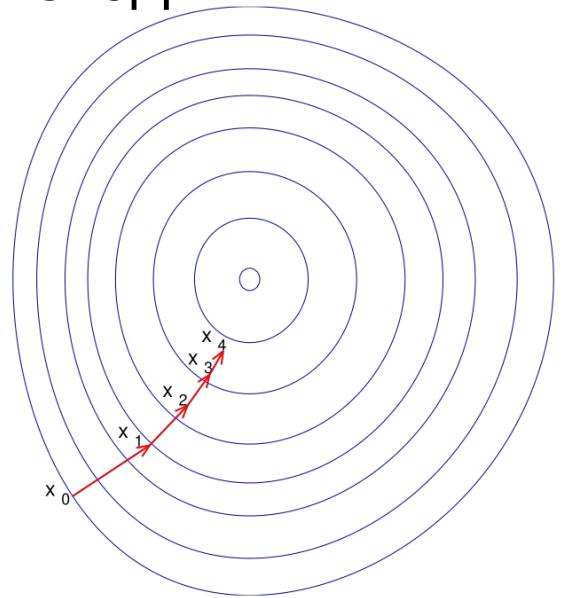
arxiv: 2010.14462

Процесс обучения. Градиентные методы

$$L(\mathbf{W}) = \sum_{i=1}^{N} l(\mathbf{y_i}, \mathbf{f}(\mathbf{W}, \mathbf{x_i})) =$$

$$= \{\text{например, если задача регрессии}\} =$$

$$= \sum_{i=1}^{N} |\mathbf{y_i}, \mathbf{f}(\mathbf{W}, \mathbf{x_i})|^2 \longrightarrow min$$



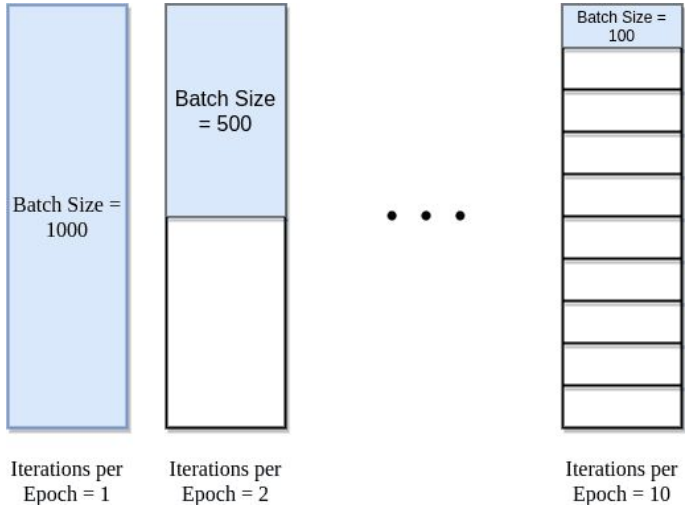
$$\mathbf{W}_k = \mathbf{W}_{k-1} - \gamma \nabla L(\mathbf{W})\big|_{W_{k-1}}$$

Процесс обучения. Стохастический градиент

$$\frac{\partial}{\partial W}\mathbb{E}\left[l(\mathbf{y_i}, f(W, \mathbf{x_i}))\right] = \frac{\partial}{\partial W}\mathbb{E}\left[\frac{1}{B}\sum_{i=1}^{B} l(\mathbf{y_i}, f(W, \mathbf{x_i}))\right] =$$

$$= \mathbb{E}\left[\frac{1}{B}\sum_{i=1}^{B} \frac{\partial}{\partial W} l(\mathbf{y_i}, f(W, \mathbf{x_i}))\right] \approx$$

$$\approx \frac{1}{B}\sum_{i=1}^{B} \frac{\partial}{\partial W} l(\mathbf{y_i}, f(W, \mathbf{x_i}))$$



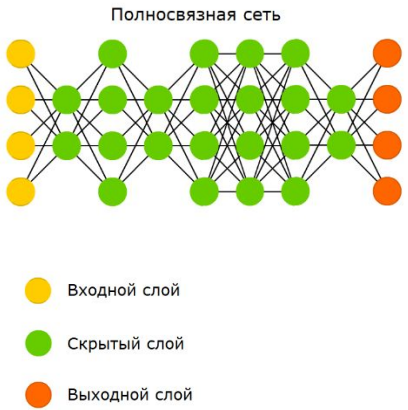
input : γ(lr), β_1, β_2(betas), θ_0(params), $f(\theta)$(objective), ϵ (epsilon)
λ(weight decay), *amsgrad*, *maximize*

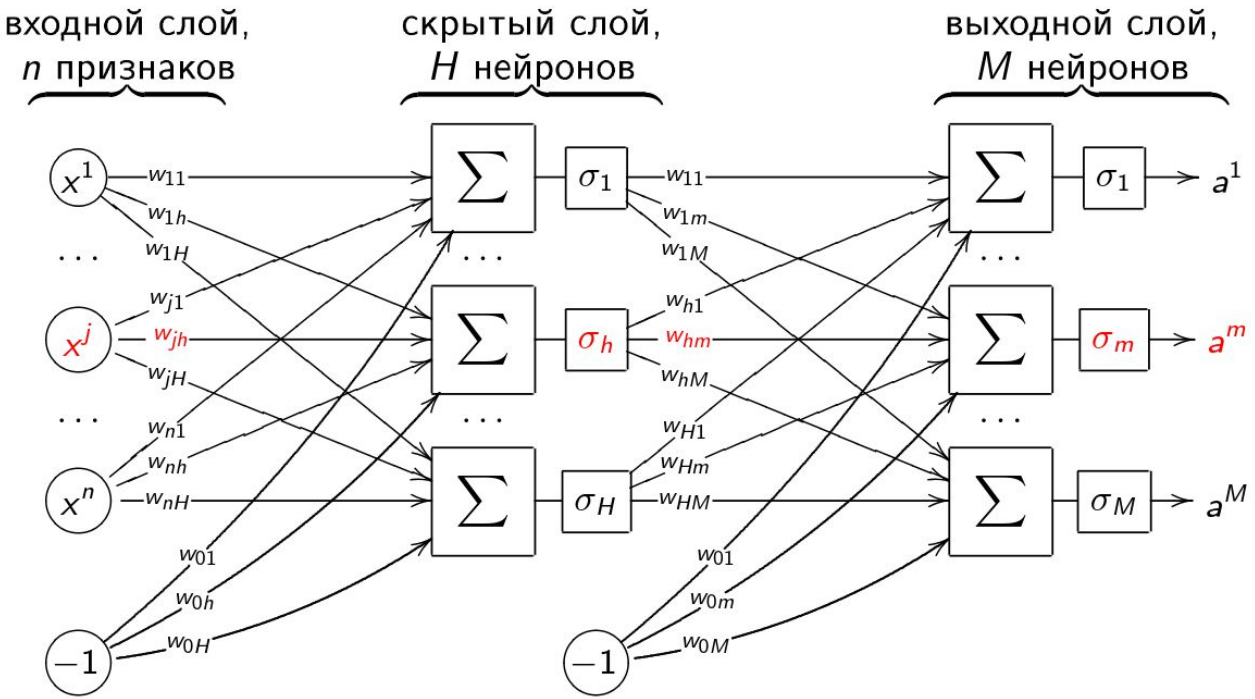
initialize : $m_0 \leftarrow 0$ (first moment), $v_0 \leftarrow 0$ (second moment), $\widehat{v_0}^{max} \leftarrow 0$

for $t = 1$ **to** ... **do**
  if *maximize* :
    $g_t \leftarrow -\nabla_\theta f_t(\theta_{t-1})$
  else
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
  $\theta_t \leftarrow \theta_{t-1} - \gamma\lambda\theta_{t-1}$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
  $\widehat{m_t} \leftarrow m_t / \left(1-\beta_1^t\right)$
  $\widehat{v_t} \leftarrow v_t / \left(1-\beta_2^t\right)$
  if *amsgrad*
    $\widehat{v_t}^{max} \leftarrow \max(\widehat{v_{t-1}}^{max}, \widehat{v_t})$
    $\theta_t \leftarrow \theta_t - \gamma\widehat{m_t} / \left(\sqrt{\widehat{v_t}^{max}} + \epsilon\right)$
  else
    $\theta_t \leftarrow \theta_t - \gamma\widehat{m_t} / \left(\sqrt{\widehat{v_t}} + \epsilon\right)$

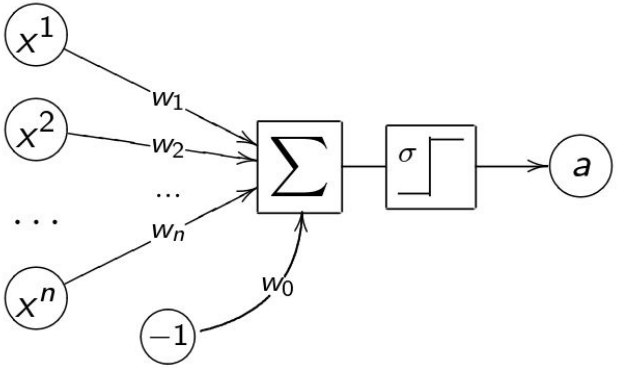
return θ_t

Объединение в слои

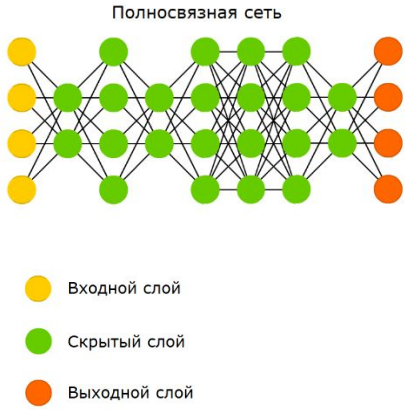




Модель нейрона.



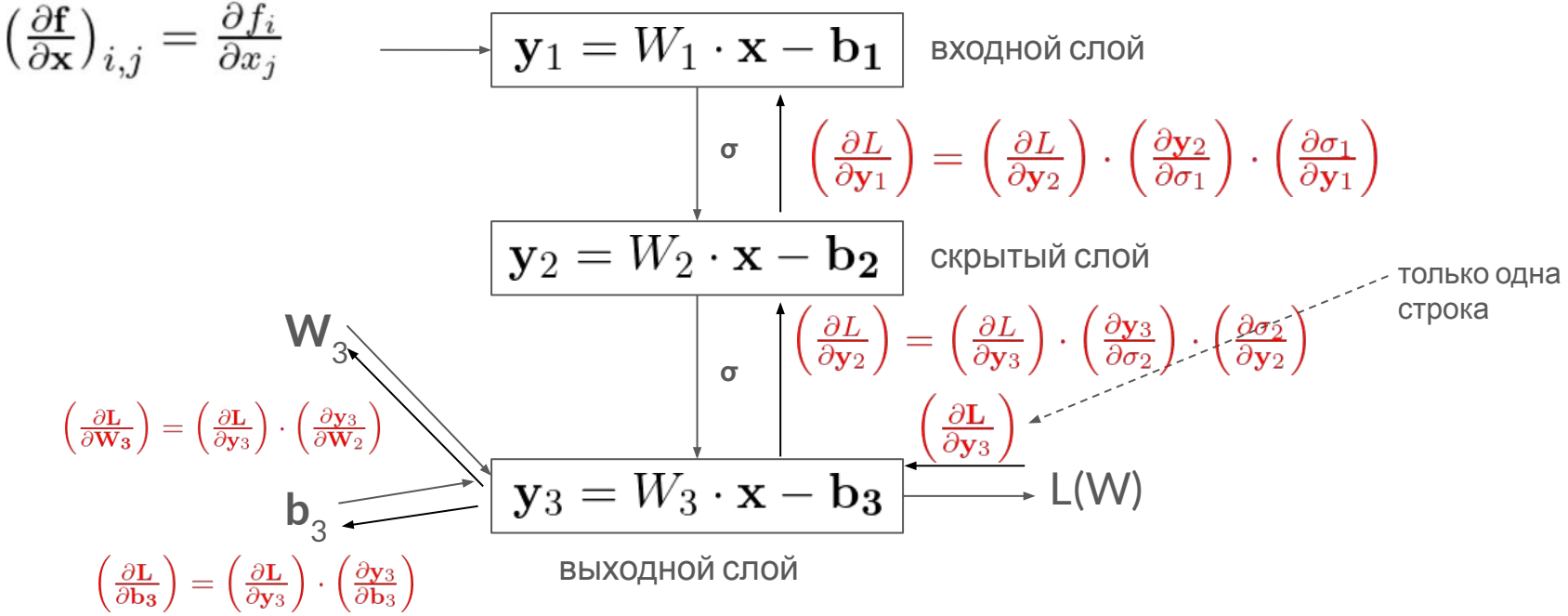
$$a(x, w) = \sigma\big(\langle w, x\rangle\big) = \sigma\left(\sum_{j=1}^{n} w_j f_j(x) - w_0\right),$$



$$layer(\mathbf{x})_k = (\sigma \circ Linear)(\mathbf{x})_k = \sigma\left(-b_k + \sum_j W_{k,j} x_j\right)$$

Как от этого считать градиент по весам?

Обратное распространение ошибки



Vector-**J**acobian (**VJP**) and **J**acobian-**V**ector (**JVP**) **p**roducts

$f : \mathbb{R}^n \to \mathbb{R}^m$ функция

$\partial f(x) \in \mathbb{R}^{m \times n}$ матрица Якоби

$(x, v) \mapsto \partial f(x) v$ JVP

$(x, v) \mapsto v \partial f(x),$ VJP

$$\left(\frac{\partial L}{\partial \mathbf{y}_1}\right) = \left(\left(\frac{\partial L}{\partial \mathbf{y}_2}\right) \cdot \left(\frac{\partial \mathbf{y}_2}{\partial \sigma_1}\right)\right) \cdot \left(\frac{\partial \sigma_1}{\partial \mathbf{y}_1}\right)$$ тоже VJP!

$$\partial(\mathbf{XY}) = (\partial \mathbf{X})\mathbf{Y} + \mathbf{X}(\partial \mathbf{Y}) \quad (37)$$

$\mathbf{v}\partial\left(W\mathbf{x} - \mathbf{b}\right) = \mathbf{v}W$ Сложность вычисления сравнима с прямым!

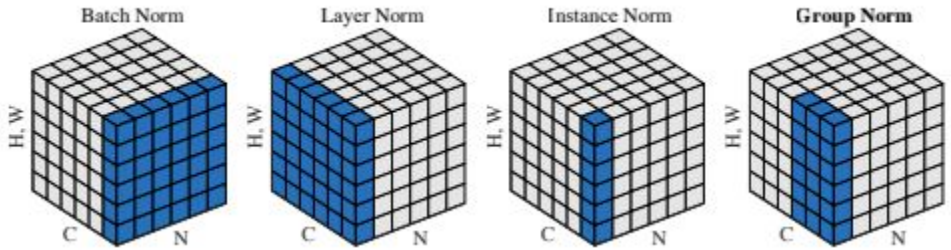
https://docs.jax.dev/en/latest/notebooks/autodiff_cookbook.html

Нормализации пакетные, по слою, …

$$\mu := \alpha\mu + (1-\alpha)\mu_{\mathcal{B}},$$
$$\sigma^2 := \alpha\sigma^2 + (1-\alpha)\sigma^2_{\mathcal{B}},$$

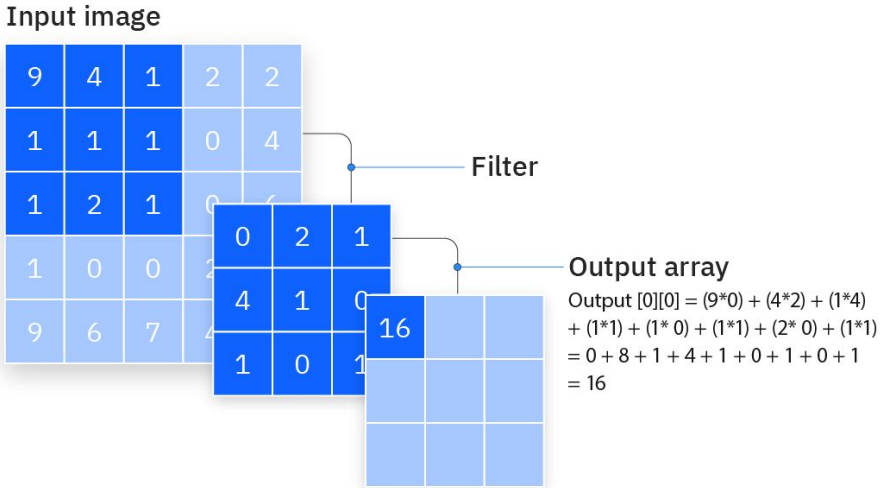
$$\hat{x} = BN(x) = \gamma\frac{x-\mu}{\sqrt{\sigma^2+\epsilon}} + \beta$$

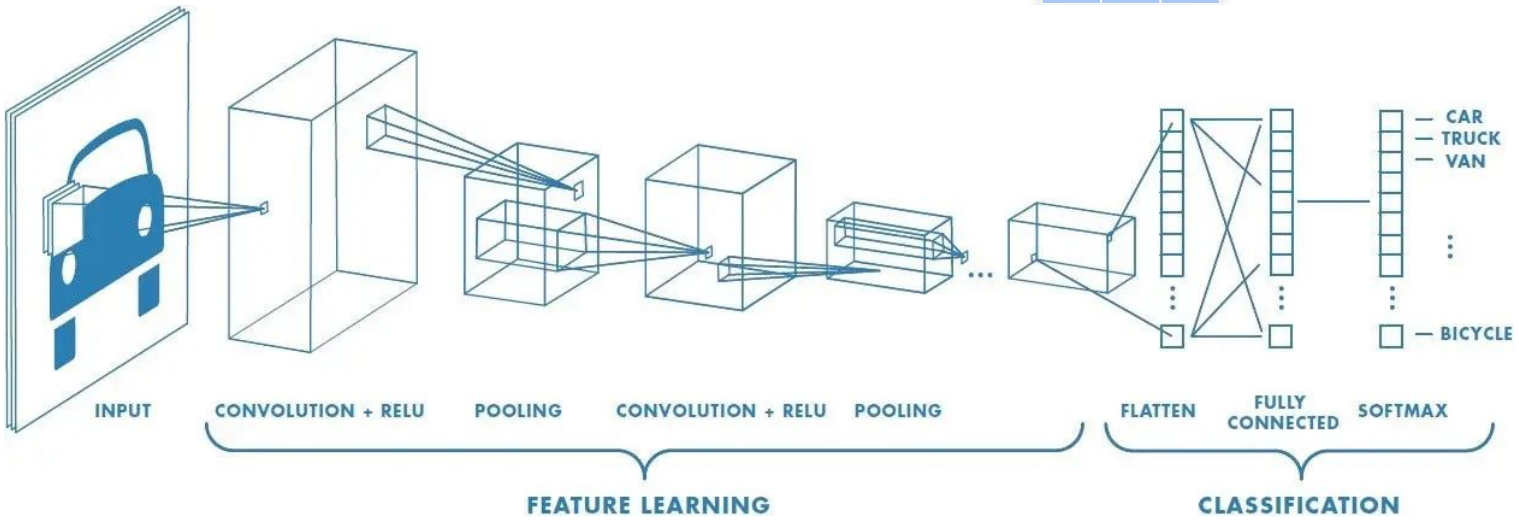
Мотивация – хотим увеличить количество слоев, но градиенты экспоненциально растут или убывают на слоях



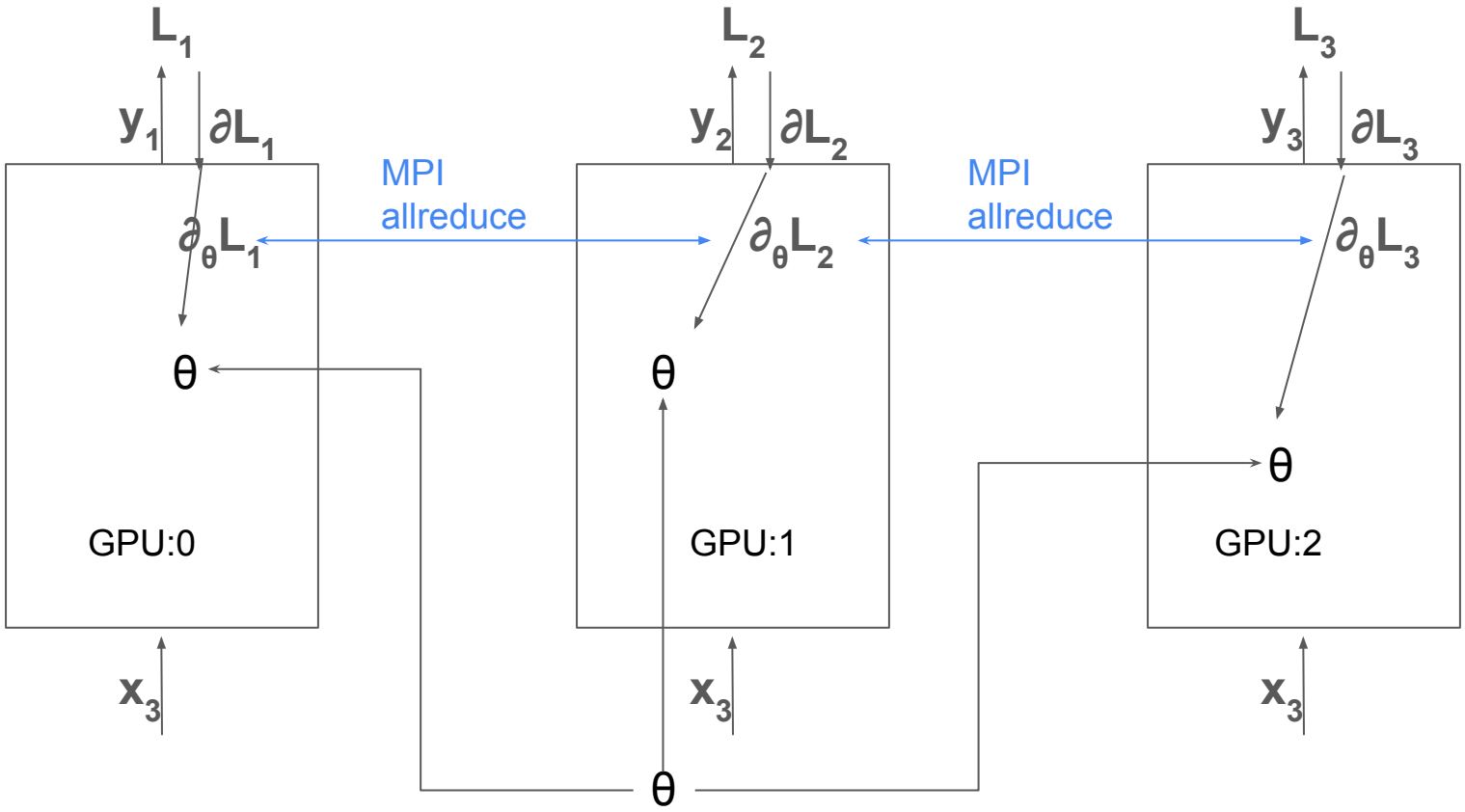
arxiv:1502.03167

Сверточные слои

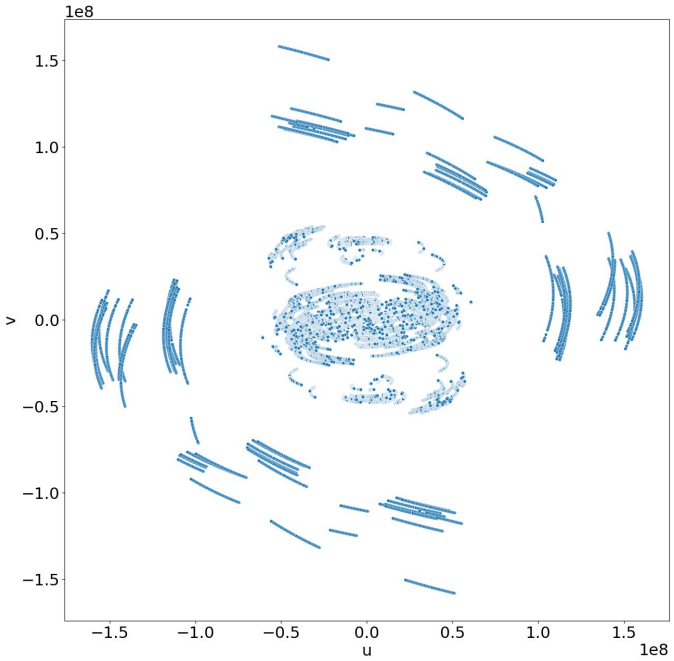




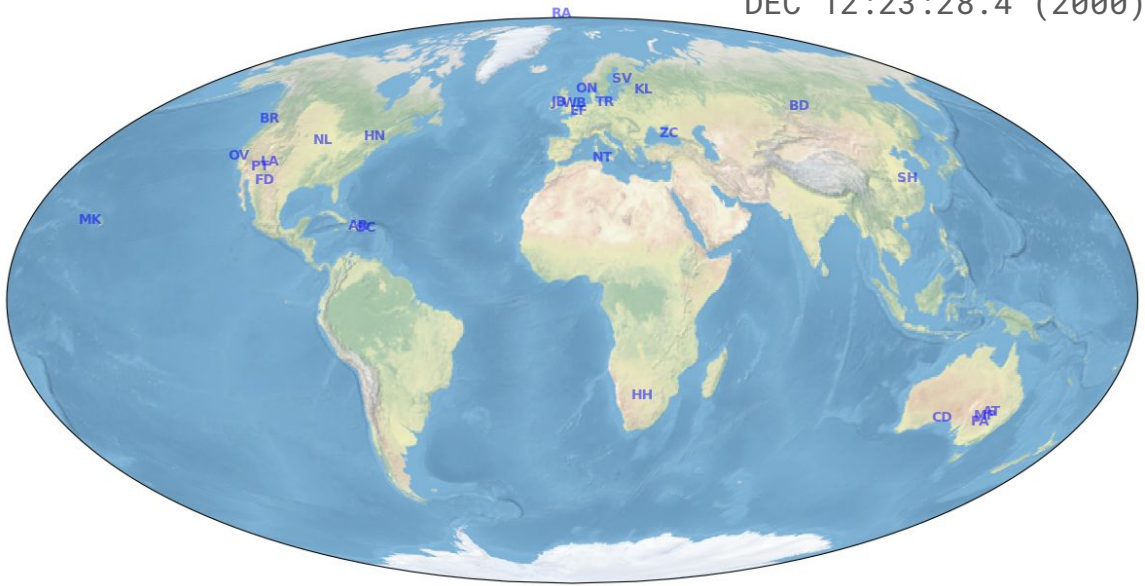
Distributed **D**ata **P**arallel



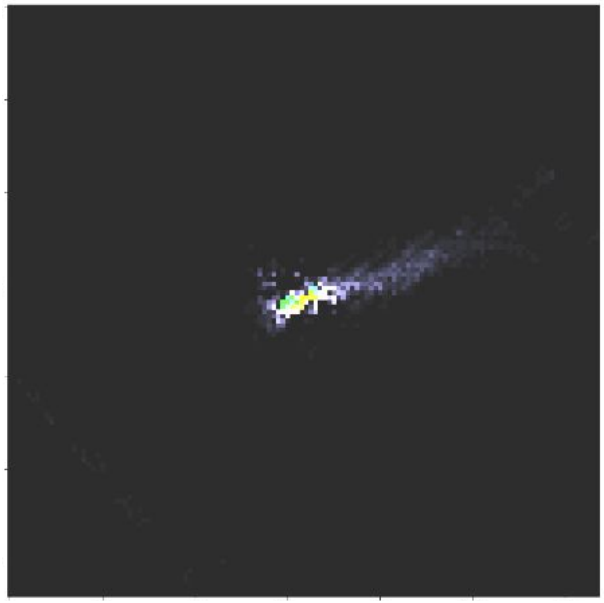
Данные наблюдений M87.

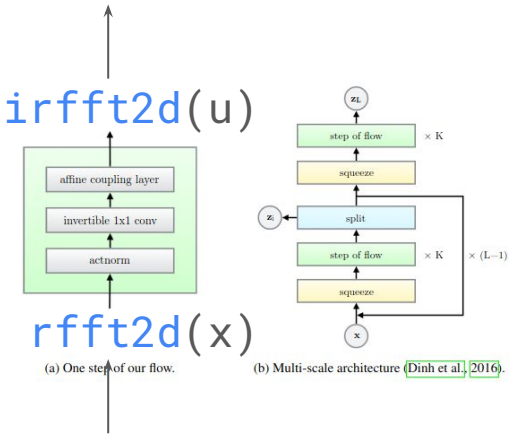


1228+126 LL-POL
1652.25 MHz
RA 12:30:49.4 (2000)
DEC 12:23:28.4 (2000)



Блок быстрого преобразования Фурье





(a) One step of our flow. (b) Multi-scale architecture (Dinh et al., 2016).

Планы. Теорема об универсальной аппроксимации для потоковых нейросетей

Neural Autoregressive Flows

Chin-Wei Huang [1 2 *] David Krueger [1 2 *] Alexandre Lacoste [2] Aaron Courvill

Abstract

Normalizing flows and autoregressive models have been successfully combined to produce state-of-the-art results in density estimation, via Masked Autoregressive Flows (MAF) (Papamakarios et al., 2017), and to accelerate state-of-the-art WaveNet-based speech synthesis to 20x faster than real-time (Oord et al., 2017), via Inverse Autoregressive Flows (IAF) (Kingma et al., 2016). We unify and generalize these approaches, replacing the (conditionally) affine univariate transformations of MAF/IAF with a more general class of invertible univariate transformations expressed as monotonic neural networks. We demonstrate that the proposed **neural autoregressive flows (NAF)** are universal approximators for continuous probability distributions, and their greater expressivity allows them to better capture multimodal target distributions. Experimentally, NAF yields state-of-the-art performance on a suite of density estimation tasks and outperforms IAF in variational autoencoders trained on binarized MNIST. [1]

1. Introduction

implicit losses, in the case of adversar
low et al., 2014). (2) In the context of v
(Rezende & Mohamed, 2015), they ca
the variational approximation to the pc
izing more complex distributions. Th
a poor variational approximation to t
to reflect the right amount of *uncertai*
(Turner & Sahani, 2011), resulting i
reliable predictions. We are thus inte
techniques for normalizing flows.

Recent work by Kingma et al. (2016
gressive models as invertible transfo
constructing normalizing flows. The ir
process, unlike sampling from the aut
not sequential and thus can be acceler
putation. This allows multiple layers
be stacked, increasing expressiveness
inference (Kingma et al., 2016) or bett
for generative models (Papamakarios e
also makes it possible to improve on
tional factorization assumed by autoregressive models such as PixelRNN or PixelCNN (Oord et al., 2016), and thus define a more flexible joint probability.

We note that the normalizing flow introduced by Kingma et al. (2016) only applies an affine transformation of each scalar random variable. Although this transformation is

Theorem 1. (DSF universally transforms any random variables into any desired random variables) *Let X be a random vector in an open set $\mathcal{U} \subset \mathbb{R}^m$. Let Y be a random vector in $\mathbb{R}^m$. Assume both X and Y have a positive and continuous probability density distribution. Then there exists a sequence of functions $(K_n)_{n\geq 1}$ parameterized by autoregressive neural networks in the following form*

$$K(\mathbf{x})_t = \sigma^{-1}\left(\mathcal{S}\left(x_t; \mathcal{C}_t(x_{1:t-1})\right)\right) \qquad (14)$$

where $\mathcal{C}_t = (a_{tj}, b_{tj}, \tau_{tj})_{j=1}^n$ are functions of $x_{1:t-1}$, such that $Y_n \doteq K_n(X)$ converges in distribution to Y.