 doi:10.20944/preprints202209.0124.v1

Article

Accelerating Extreme Search of Multidimensional Functions Based on Natural Gradient Descent with Dirichlet Distributions

Ruslan Abdulkadirov [1]*, Pavel Lyakhov [1,2], Nikolay Nagornov [2]

[1] North-Caucasus Center for Mathematical Research, North-Caucasus Federal University, Stavropol, 355009, Russian Federation
[2] Department of Mathematical Modeling, North-Caucasus Federal University, Stavropol, 355009, Russian Federation
* Correspondence: ruslanabdulkadirovstavropol@gmail.com

Abstract: In this work, we explore the extreme searching of multidimensional functions by natural gradient descent based on Dirichlet and generalized Dirichlet distributions. The natural gradient is based on describing multidimensional surface with probability distributions, which allows us to reduce changing the accuracy of gradient and step-size. In this article, we propose an algorithm of natural gradient descent based on Dirichlet and generalized Dirichlet distributions. We demonstrate that the natural gradient descent with step-size adaptation with Dirichlet and generalized Dirichlet distributions has higher accuracy and does not take a large number of iterations for minimizing test functions than gradient descent and Adam.

Keywords: Natural gradient descent, optimization, K-L divergence, Dirichlet distribution, generalized Dirichlet distribution

1. Introduction

The optimization methods remain the most important and actual problem in artificial neural networks, which significantly impact the process of recognition. It lets us solve many problems approximately without a complex analytical approach. The most usable optimization algorithm in machine learning is stochastic gradient descent (SGD) and its modifications such as AdaGrad in [1] and [2], RMSprop in [3], ADADELTA and Adam algorithm in [4] and [5], respectively. But they are not rapid enough and, usually, converge to thelocal extreme. Even step-size adaptation from [6] can't obtain the required accuracy in minimum time. But researching the loss function from the geometrical point of view, especially Riemannian, can perform the minimization.

The metric properties are described in Riemannian geometry on arbitrary n-dimensional smooth manifolds with local coordinates. According to the kind of manifold, we can provide the gradient flow that improves the quality of the optimization process. The gradient flow from [7] is the product between metric tensor and gradient of the optimizing function. The optimization accelerates the computations and minimizes iterations (epochs) applying smooth manifolds. But in this manuscript, we provide the extreme search with manifolds of probability distributions.

Probability distribution manifolds mostly meat in information geometry, where the analog of gradient flow is natural gradient. The natural gradient in information geometry is the product between Fisher information matrix and gradient of the optimizing function. The Fisher matrix is calculated by the Kullback–Leibler divergence (K–L divergence in [8]–[10]). Remarkable that changing the length of step or value of the gradient for NGD is not necessary for increasing the accuracy. Then selecting appropriate parameters for distributions suffices.

Natural gradient descent (NGD) is an alternative for stochastic gradient descent and its modifications, as it was noted in [11]. Unfortunately, for models with many parameters such

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 September 2022 doi:10.20944/preprints202209.0124.v1

as large neural networks, computing the natural gradient is impractical due to the extreme size of the Fisher matrix. That problem can be solved, using various approximations to the Fisher matrix in [12] and [13]. It is designed to facilitate the computation, storing, and finally inverting the exact Fisher.

In this article, we propose algorithm of natural gradient descent based on Dirichlet and generalized Dirichlet distributions. We demonstrate that the natural gradient descent with step-size adaptation with Dirichlet and generalized Dirichlet distributions has higher accuracy and does not take a large number of iterations for minimizing test functions than gradient descent and Adam. Later we discuss results, perspectives, and directions of the developing new modifications for natural gradient descent.

The remaining of the paper is organized as follows. Section II presents the background of gradient descent, Adam, and gradient flow. Section III demonstrates calculations of Fisher matrices with Dirichlet and generalized Dirichlet distributions and proposing the Algorithm 3. Section IV represents experiments of minimization with graphs and tables. In section V reported conclusions and suggestions for developing the natural gradient descent further.

2. Preliminaries

2.1. Gradient Descent with Step-Size Adaptation

Let's consider optimizing a smooth function $f : \Omega \to \mathbb{R}$ over closed convex set $\Omega \in \mathbb{R}$. The problem of minimization is to calculate $\min_{x\in\Omega}(f(x))$. This is necessary part of every artificial neural network.

The gradient descent with appropriate step-size adaptation in [6] has advantages in rate and accuracy over stochastic gradient descent.

Gradient descent with step-size adaptation is defined as Algorithm 1.

Algorithm 1 Gradient Descent with Step-Size Adaptation

Input: $x \in \mathbb{R}^n$ (starting point), $f(x)$ (scalar function), $\nabla f(x)$ (gradient), a (initial step-size)
Output: some x minimizing f

initialize $f_x = f(x) \in \mathbb{R}$
$g = \nabla f(x)^T \in \mathbb{R}^n$ and Fisher matrix F
for i from 0 to n - 1 **do**
  $y \leftarrow x - ag/|g|$
  $f_y \leftarrow f(y)$
  if $f_y < f_x$ **then**
    $x \leftarrow y$
    $f_x \leftarrow f_y$
    $g \leftarrow \nabla f(x)^T$
    $a \leftarrow 1.2a$
  else
    $a \leftarrow 0.5a$
  end if
end for

Remark that in general cases gradient descent can not reach the minimum, because of constant step and confusing of gradient in case of several local extremes. Submitting step-size adaptation does not guarantee descent into global minimum, due to big number of local minimums. But it allows us to increase the accuracy. This problem led the researchers to replace this method with Adam.

2.2. Adam Algorithm

The Adam algorithm ([5]) is an attempt to improve the stochastic gradient descent, which updates exponential moving averages of the gradient m_t and the squared gradient v_t with the hyper-parameters $\beta_1, \beta_2 \in [0, 1)$ control the exponential decay rates of these moving averages. The moving averages themselves are estimates of the first moment (the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 September 2022 doi:10.20944/preprints202209.0124.v1

mean) and the second raw moment (the uncentered variance) of the gradient. However, these moving averages are initialized as (vectors of) 0's, leading to moment estimates that are biased towards zero, especially during the initial time steps, and especially when the decay rates are small. The good news is that this initialization bias can be easily counteracted, resulting in bias-corrected estimates $\hat{m}_t$ and $\hat{v}_t$. The step-size adaptation improves the quality of the Adam algorithm, which means accelerating the minimization and increasing the accuracy. These amendments can be helpful in deep neural networks because the optimizer gives more accurate results in less time.

Let us present the pseudo-code of the Adam method in Algorithm 2.

Algorithm 2 Adam algorithm

Input: $x \in \mathbb{R}^n$ (starting point), $f(x)$ (scalar function), $\nabla f(x)$ (gradient), a (initial step-size), β_1, β_2 (exponential decay rates)
Output: some x minimizing f

initialize $f_x = f(x) \in \mathbb{R}, g = \nabla f(x)^T \in \mathbb{R}^n, m_0 = 0, v_0 = 0$
for i from 0 to n - 1 **do**
  $g_i \leftarrow \nabla f(x_i)$
  $m_i \leftarrow \beta_1 m_{i-1} + (1-\beta_1) g_i$
  $v_i \leftarrow \beta_2 v_{i-1} + (1-\beta_2) g_i^2$
  $\hat{m}_i \leftarrow m_i / (1-\beta_1^i)$
  $\hat{v}_i \leftarrow v_i / (1-\beta_1^i)$
  $y \leftarrow x - a \cdot \hat{m}_i / (\sqrt{\hat{v}_i} + \epsilon)$
  if $f_y < f_x$ **then**
    $x \leftarrow y$
    $f_x \leftarrow f_y$
    $a \leftarrow 1.2a$
  else
    $a \leftarrow 0.5a$
  end if
end for

In process of learning neural networks, the Adam algorithm is the most preferred optimization method, because it converges faster and gives the required accuracy. But this algorithm does not contain the engaging the curvature of the function $n-1$-surfaces, for $n \geq 2$. Therefore it does not reach the global minimum for little steps in the case of very convex functions such as Rastrigin, which is shown in subsection 4.2.

2.3. Background on Riemannian Gradient Flow

The main idea of natural gradient descent initially comes from Riemannian geometry, where the definitions of derivative, flow, and curvature are generally described.

Let $(\mathcal{M}, g)$ be an Riemannian manifold, where the topological space $M = \mathbb{R}^n$ and metric tensor $g : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$.

The tangent space T_x, for which holds $T_x\mathcal{M} = \mathbb{R}^n$, and the metric tensors $g(x) \in \mathbb{S}^n_{++}$, where $\mathbb{S}^n_{++}$ is the cone of real symmetric definite positive matrices [14], can be taken for manifold $\mathcal{M}$. The tensor matrix $g(x, x+\delta x)$ defines the local distances at x as $d(x, x+\delta x)^2 = \delta x^T g(x, \delta x)\delta x$ for $\delta x \to \infty$.

We denote by

$$\nabla_g f \mid_\Omega = g(x, x+\delta x)^{-1} \nabla f(x) \tag{1}$$

the corresponding Riemannian gradient vector field of the objective function f restricted to Ω [14].

Information geometry [15] is concerned with a manifold of probability distributions, e.g. in a parametric family $p(x;\theta) : \theta \in \Theta \subseteq \mathbb{R}^n$, typically endowed with the metric derived from the Kullback-Leibler divergence. Natural gradient is defined in such manifolds.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 September 2022 doi:10.20944/preprints202209.0124.v1

3. Theoretical calculations

3.1. Natural Gradient Descent and K-L divergence

Natural Gradient Descent in [11] is obtained as the forward Euler discretization with stepsize η of the gradient flow (1):

$$x^{(k+1)} = x^{(k)} - \eta_k F(x^{(k)})^{-1} \nabla f(x^{(k)}), \tag{2}$$

where $x^{(0)} = x_0$.

The main part of natural gradient descent is a Fisher matrix $F(x^k)$ from [11], that can be calculated on manifold of probability distributions. Suppose we optimize the $f(\theta)$. Let $p(x;\theta)$ is some family of probability distributions over x parametrized by a vector of real numbers θ. Let's the KL-divergence be

$$KL(p(x;\theta_t)||p(x;\theta_t+\delta\theta)) = \int p(x;\theta_t)\log\frac{p(x;\theta_t)}{p(x;\theta_t+\delta\theta)}dx$$
$$= \int p(x;\theta_t)\log p(x;\theta_t)dx - \int p(x;\theta_t)\log p(x;\theta_t+\delta\theta)dx. \tag{3}$$

The second Taylor series expansion of the function f is

$$f(\theta) \approx f(\theta_t) + \nabla f(\theta_t)^T\delta\theta + \frac{1}{2}\delta\theta^T\nabla^2 f(\theta_t)\delta\theta, \tag{4}$$

where $\theta = \theta_t + \delta\theta$, and $\nabla^2 f = Hess(f)$ is an Hessian matrix.

Substituting the K-L divergence (3) into second Taylor series (4), we receive the following expansion

$$KL(p(x;\theta_t)||p(x;\theta_t+\delta\theta)) \approx \int p(x;\theta_t)\log p(x;\theta_t)dx$$
$$-\int p(x;\theta_t)\left[\log p(x;\theta_t) + \left(\frac{\nabla p(x;\theta_t)}{p(x;\theta_t)}\right)^T\delta\theta\right]dx + \frac{1}{2}\int p(x;\theta_t)\left[\delta\theta^T\left(\nabla^2\log p(x;\theta_t)\right)\delta\theta\right]dx$$
$$= \int p(x;\theta_t)\log\frac{p(x;\theta_t)}{p(x;\theta_t)}dx - \left(\int\nabla p(x;\theta_t)dx\right)^T\delta\theta - \frac{1}{2}\delta\theta^T\left(\int p(x;\theta_t)\nabla^2\log p(x;\theta_t)dx\right)\delta\theta.$$

First two integrals are equal to 0, because $\log\frac{p(x;\theta_t)}{p(x;\theta_t)} = \log 1 = 0$ and

$$\int\nabla p(x;\theta_t)dx = \nabla\int p(x;\theta_t)dx = \nabla 1 = 0.$$

Therefore we receive K-L divergence for continuous probability distribution.

$$KL(p(x;\theta_t)||p(x;\theta_t+\delta\theta)) \approx \frac{1}{2}\delta\theta^T\left(\int p(x;\theta_t)\nabla^2\log p(x;\theta_t)dx\right)\delta\theta.$$

Next we can provide the following Hessian $\nabla^2 log\ p(x;\theta_t)$, then it holds that

$$\nabla^2\log p(x;\theta_t) = \nabla\log p(x;\theta_t)\nabla\log p(x;\theta_t))^T.$$

Then the K-L divergence can be represented as

$$KL(p(x;\theta_t)||p(x;\theta_t+\delta\theta)) = -\frac{1}{2}\delta\theta^T\mathbb{E}\left[\nabla\log p(x;\theta_t)\nabla\log p(x;\theta_t)^T\right]\delta\theta, \tag{5}$$

where $-\mathbb{E}\left[\nabla\log p(x;\theta_t)\nabla\log p(x;\theta_t)^T\right] = F(\theta_t)$ is a Fisher information matrix, which is a Riemannian structure on manifold of probability distributions.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 September 2022 doi:10.20944/preprints202209.0124.v1

3.2. Fisher matrix for Dirichlet and Generalized Dirichlet Distributions

The Dirichlet distribution of order $K \geq 2$ with parameters $\alpha_1, ..., \alpha_K > 0$ [16] has a probability density function with respect to Lebesgue measure on the Euclidean space $\mathbb{R}^{K-1}$ given by

$$f(x_1, ..., x_K; \alpha_1, ..., \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}, \; B(\alpha) = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)}, \tag{6}$$

where $\{x_i\}_{i=1}^{K}$ belongs to $K-1$ simplex.

Now let's calculate the logarithm of Dirichlet distribution.

$$\log f(x_1, ..., x_K; \alpha_1, ..., \alpha_K) = \log \left[\frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \right]$$
$$= \log \Gamma(\sum_{i=1}^{K} \alpha_i) - \sum_{i=1}^{K} \log \Gamma(\alpha_i) + \sum_{i=1}^{K} (\alpha_i - 1) \log x_i.$$

Second order partial derivative of f with respect to α:

$$\frac{\partial^2}{\partial a_j \partial a_k} \log f = \psi' \left(\sum_{i=1}^{K} \alpha_i \right), \; \frac{\partial^2}{\partial a_j^2} \log f = \psi' \left(\sum_{i=1}^{K} \alpha_i \right) - \psi'(\alpha_j).$$

Therefore, we can find the Fisher matrix

$$F_{Dir}(\alpha) = \begin{pmatrix} \psi'(\alpha_1) - \psi'(\sum_i \alpha_i) & ... & -\psi'(\sum_i \alpha_i) \\ ... & ... & ... \\ -\psi'(\sum_i \alpha_i) & ... & \psi'(\alpha_K) - \psi'(\sum_i \alpha_i) \end{pmatrix}. \tag{7}$$

The generalized Dirichlet distribution [**?**] for $x_1 + ... + x_K \leq 1$ and $\alpha_i > 0, \beta_i > 0, \; i = 1, ..., K-1$ has a probability density function, which is defined as

$$f(x_1, ..., x_K; \alpha_1, ..., \alpha_K, \beta_1, ..., \beta_K) = \prod_{i=1}^{K} \frac{1}{B(\alpha_i, \beta_i)} x_i^{\alpha_i - 1} \left(1 - \sum_{j=1}^{i} x_j \right)^{\gamma_i}, \tag{8}$$

where $\gamma_i = \beta_i - \alpha_{i+1} - \beta_{i+1}$ for $i = 1, ..., K-1$ and $\gamma_K = \beta_{K-1}$.

The logarithm is

$$\log f = \log \left[\prod_{i=1}^{K} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} x_i^{\alpha_i - 1} \left(1 - \sum_{j=1}^{i} x_j \right)^{\gamma_i} \right] = \sum_{i=1}^{K} \log \Gamma(\alpha_i + \beta_i)$$
$$- \sum_{i=1}^{K} \log \Gamma(\alpha_i) - \sum_{i=1}^{K} \log \Gamma(\beta_i) + \sum_{i=1}^{K} (\alpha_i - 1) \log x_i + \sum_{i=1}^{K} \gamma_i \log(1 - \sum_{j=1}^{i} x_j).$$

The second order partial derivatives of $\log f(x; \alpha, \beta)$:

$$1) \; \frac{\partial^2}{\partial \alpha_j \partial \alpha_l} \log f = \frac{\partial^2}{\partial \beta_j \partial \beta_l} \log f = \frac{\partial^2}{\partial \alpha_j \partial \beta_l} \log f = 0, \; j \neq l,$$
$$2) \; \frac{\partial^2}{\partial \alpha_j^2} \log f = \psi'(\alpha_j + \beta_j) - \psi'(\alpha_j), \; \frac{\partial^2}{\partial \beta_j^2} \log f = \psi'(\alpha_j + \beta_j) - \psi'(\beta_j),$$
$$3) \; \frac{\partial^2}{\partial \alpha_j \partial \beta_j} \log f = \frac{\partial^2}{\partial \beta_j \partial \alpha_j} \log f = \psi'(\alpha_j + \beta_j).$$

Then the Fisher matrix for generalized Dirichlet distribution has the following form:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 September 2022 doi:10.20944/preprints202209.0124.v1

$$F_{GenDir}(\alpha) = \begin{pmatrix} \Psi_1 & \mathcal{O} & ... & \mathcal{O} \\ \mathcal{O} & \Psi_2 & ... & ... \\ ... & ... & ... & ... \\ \mathcal{O} & ... & \mathcal{O} & \Psi_K \end{pmatrix}, \tag{9}$$

where

$$\Psi_i = \begin{pmatrix} \psi'(\alpha_i) - \psi'(\alpha_i + \beta_i) & -\psi'(\alpha_i + \beta_i) \\ -\psi'(\alpha_i + \beta_i) & \psi'(\beta_i) - \psi'(\alpha_i + \beta_i) \end{pmatrix} \textit{ and } \mathcal{O} \textit{ is zero matrix.}$$

According to Fisher information matrix for Dirichlet and generalized Dirichlet distributions and adding the step-size adaptation, we propose Algorithm 3.

Algorithm 3 Natural Gradient Descent with Dirichlet and generalized Dirichlet distribution

Input: $x \in \mathbb{R}^n$ (starting point), $f(x)$ (scalar function), $\nabla f(x)$ (gradient), a (initial step-size)
Output: some x minimizing f

initialize $f_x = f(x) \in \mathbb{R}$, $g = \nabla f(x)^T \in \mathbb{R}^n$ and Fisher matrix F
for i from 0 to n - 1 **do**
  $y \leftarrow x - a\, F^{-1}\, g/|g|$, $f_y \leftarrow f(y)$
  if $f_y < f_x$ **then**
    $x \leftarrow y$, $f_x \leftarrow f_y$, $g \leftarrow \nabla f(x)^T$, $a \leftarrow 1.2a$
  else
    $a \leftarrow 0.5a$
  end if
end for

Remark that in Algorithm 3 it is unnecessary decreasing the length of steps or numerical value of gradient for improving final values of extremes. Fisher matrix contains parameters without items of vector x, which allows avoiding additional computations in the loop. Including curvature properties by Fisher matrix natural gradient achieve extreme faster. Finally, Fisher information matrix with generalized Dirichlet distribution is useful only in cases of 2n-dimensional surfaces, where $n \in \mathbb{N}$.

4. Experimental Part

4.1. 4-dimensional case

The behavior of the algorithms gradient descent with stepsize adaptation and natural gradient descent of Dirichlet and general Dirichlet distributions, realized by Python 3.8.10, will be observed in experiments. We choose convex and smooth functions for solving the optimization problem.

Initial points and parameters will be defined for every function. It is made to figure out the proper distribution for every experimental function.

In the first experiment, we minimize the Rayden function, which is defined as

$$f(x) = \sum_{i=1}^{4} (\exp(x_i) - x_i), \tag{10}$$

with global minimum at $x = (0, 0, 0, 0)$, where $f(x) = 4$.

In figure 1 shown that NGD with generalized Dirichlet distributions has the fastest convergence and achieves minimal value, which is equal to $4 + 6 \times 10^{-9}$. For Dirichlet distribution the optimization is fast enough and gives the least value $4 + 1 \times 10^{-9}$. For the Adam algorithm, the minimum value is $4 + 2 \times 10^{-9}$. GD with step-size adaptation gives $4 + 2 \times 10^{-8}$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 September 2022 doi:10.20944/preprints202209.0124.v1

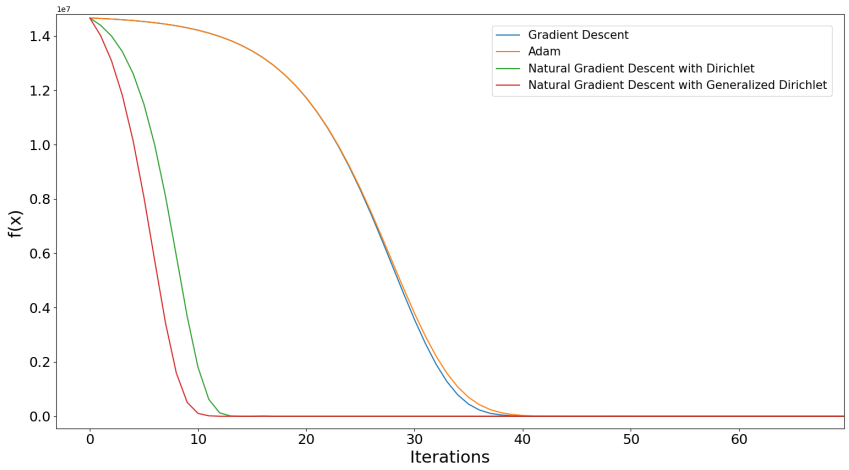


Figure 1. The rate of convergence on Rayden function using various algorithms.

The second minimization is implemented on generalized Rosenbrock function, which has the form as

$$f(x) = \sum_{i=1}^{3} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right], \tag{11}$$

where global minimum is equal to 0 at $x = (1, 1, 1, 1)$.

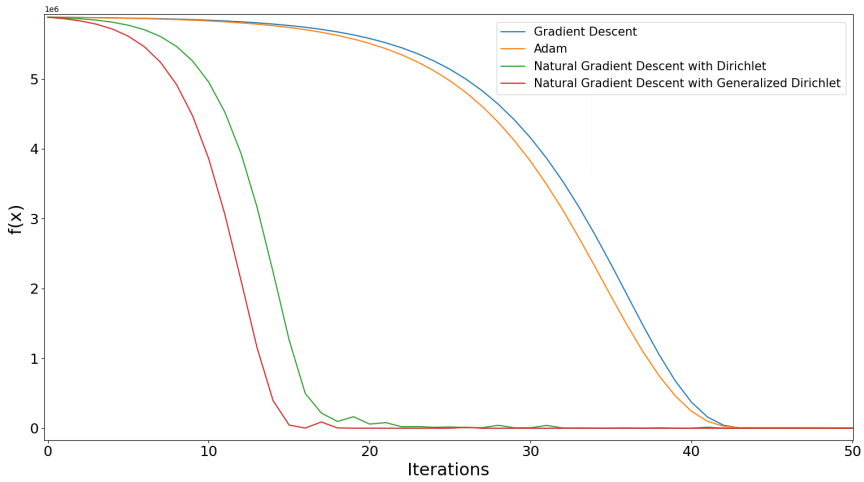


Figure 2. The rate of convergence on generalized Rosenbrock function using various algorithms.

In figure 2 shown that NGD with Dirichlet and generalized Dirichlet distributions has the fastest convergence and achieves minimal value, which is equal to 0.01095 and 0.22170, respectively. For the Adam algorithm, the minimum value is 0.01391. GD with step-size adaptation reaches 0.14325.

The extended trigonometric function is

$$f(x) = \sum_{i=1}^{4} \left[\left(4 - \sum_{j=1}^{4} \cos x_i \right) + i \cos x_i - \sin x_i \right]^2, \tag{12}$$

which has the global minimum at $x = (\pi/2, \pi/2, \pi/2, \pi/2)$, where $f(x) = 0$

In figure 3 shown that NGD with Dirichlet distributions reaches the minimum at 1.57389×10^{-10}. For Adam algorithm minimal value is 2.24709×10^{-09}. GD with step-size adaptation shows 5.70724×10^{-9}. NGD with Generalized Dirichlet distribution descended to 1.50210×10^{-09}.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 September 2022 doi:10.20944/preprints202209.0124.v1

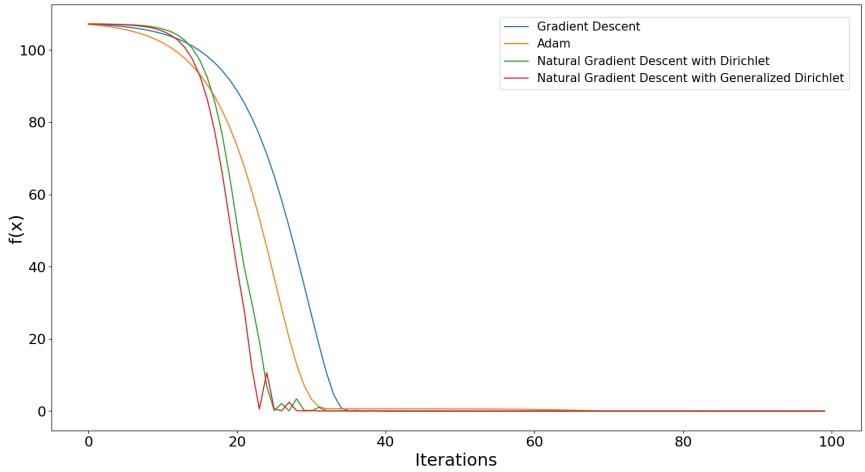


Figure 3. The rate of convergence on extended trigonometric function using various algorithms.

As we can see, the Fisher information matrix accelerates the convergence, which allows the optimizer in neural networks to work faster.

4.2. 3-dimensional case

In 3-dimensional space, we can provide the graph of convergence and descent-trajectory for each optimization method. The gradient descent and Adam algorithms with step-size adaptation remain unchanged, except the dimension of variable x. But Dirichlet and generalized Dirichlet distributions reduce to Beta distribution.

$$\mathcal{B}(x;a,b) = \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1}, \tag{13}$$

where

$$B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}dt = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)},$$

for $0 < x < 1$ and $a > 0,\ b > 0$.

The beta distribution is the Dirichlet and generalized Dirichlet distribution in 3-dimensional Euclidean space. Hence the Fisher matrix of Beta distribution is

$$F_{Beta}(a,b) = \begin{pmatrix} \psi'(a) - \psi'(a+b) & -\psi'(a+b) \\ -\psi'(a+b) & \psi'(b) - \psi'(a+b) \end{pmatrix}. \tag{14}$$

In case of 2-dimensional surfaces we can observe their graphs and descent trajectories of each optimization methods. It allows us to understand the work of every algorithm and estimate the efficiency of their models.

The surface, which is described as

$$f(x,y) = \sin(\frac{1}{2}x^2 - \frac{1}{4}y^2 + 3)\cos(2x + 1 - e^y) \tag{15}$$

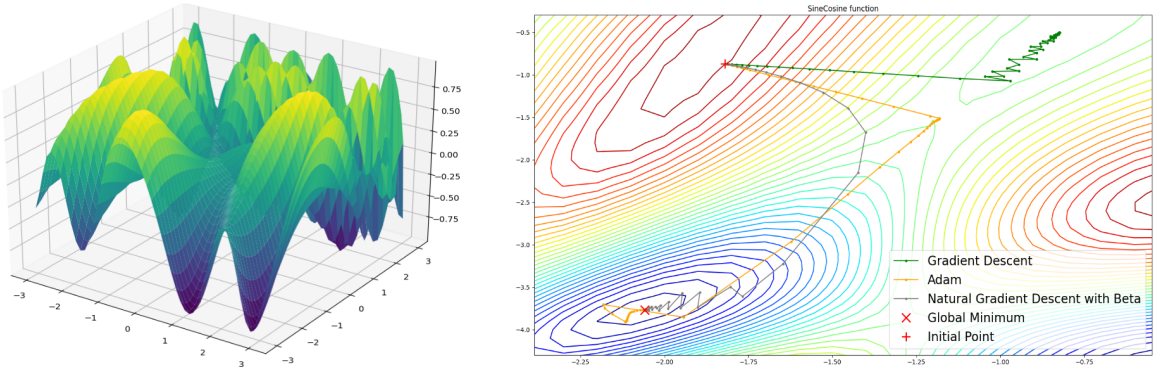
is Sine-Cosine function with initial point $(x,y) = (5,5)$.

The best result gives beta distribution and achieves $-1 + 2 \times 10^{-8}$. The Adam shows $-1 + 6 \times 10^{-8}$, but it converges slower. The gradient descent moves in the wrong direction and achieves -0.04198.

The second simulation was implemented on Rastrigin function

$$f(x) = An + \sum_{i=1}^{n}[x_i - A\cos(2\pi x_i)], \tag{16}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 September 2022 doi:10.20944/preprints202209.0124.v1

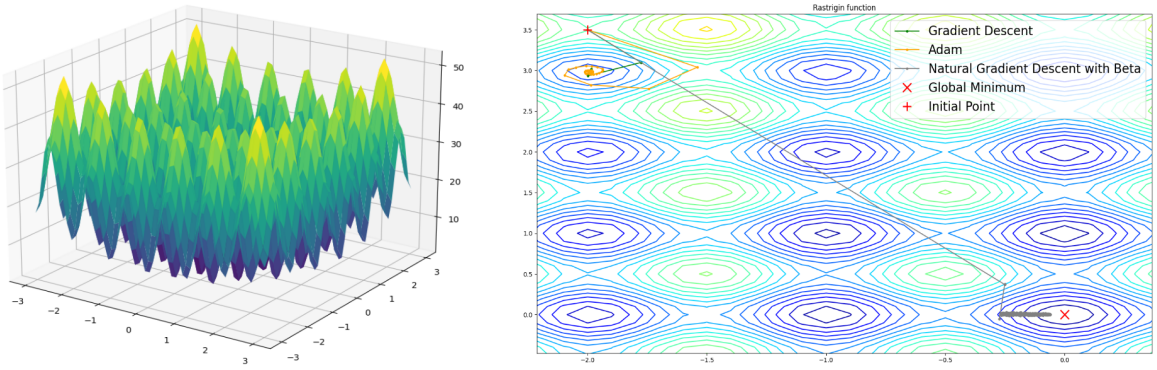


(a) (b)

Figure 4. Experiment on Sine-Cosine function: a) the appearance of the function; b) the trajectory of movement to a minimum using various algorithms.

where $A = 10$ and $x_i \in [-5.12, 5.12]$. It has the global minimum at $x = (0, 0)$, where $f(x) = 0$.

This function contains many local minimums, then the method like gradient descent will not achieve the global minimum with a small step-size. For Adam, the step-size needs to be greater than 1.6. But for natural gradient descent with beta distribution step-size can be less than 0.5.



(a) (b)

Figure 5. Experiment on Rastrigin function: a) the appearance of the function; b) the trajectory of movement to a minimum using various algorithms.

The NGD with beta distribution reached the global minimum and gave 0.69984. The Adam and GD achieved the local minimums, which does not suffice for minimization.

The third simulation was implemented on Rosenbrock function

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]. \tag{17}$$

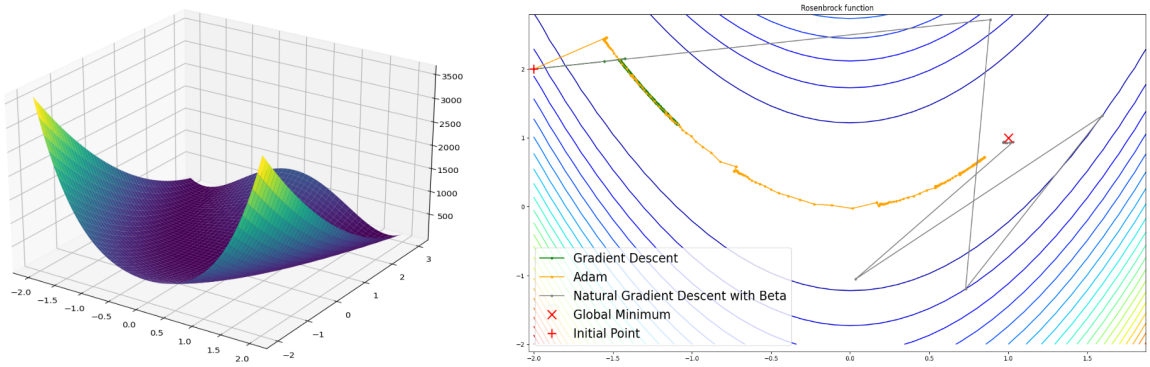
It has a global minimum at $x = (1, 1)$, where $f(x) = 0$.

In this case will be demonstrated descent in the area of local minimums, where for each method required the achieving the global minimum.

The NGD with beta distribution for the least number of iterations achieved the minimum 0.00082. The GD moves along the area of local minimums, but because of the insufficient number of iterations stops with value 4.36387. Adam goes the same way as GD does and reaches the value 0.02243, which is not progressive compared with NGD.

Let us summarize the results in the table below.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 September 2022 doi:10.20944/preprints202209.0124.v1



(a) (b)

Figure 6. Experiment on Rosenbrock function: a) the appearance of the function; b) the trajectory of movement to a minimum using various algorithms.

Table 1. Minimum values achieved by various algorithms.

Function	Optimization algorithms		
	GD [6]	Adam [5]	Proposed
Sine-Cosine	-0.04198	$-1+6\times10^{-8}$	$-1+2\times10^{-8}$
Rastrigin	12.93446	12.93451	0.69984
Rosenbrock	4.36387	0.02243	0.00082

Table 2. Number of iterations by various algorithms.

Function	Optimization algorithms		
	GD [6]	Adam [5]	Proposed
Sine-Cosine	19	100	26
Rastrigin	5	12	32
Rosenbrock	> 500	200	20

According to the results of graphs in figures 4(b) – 6(b), we can put the number of iterations into the table. We can conclude that Natural gradient descent with Dirichlet (Beta) distribution works better than known analogs. It is simple for the program realization and does for the least number of iterations can give the best results in the optimization process.

5. Discussion

According to the results of experiments, we conclude that Adam algorithm has no difficulties with reaching the minimal point with required accuracy and takes a not large number of iterations for functions with only one minimum. But in the case of functions, which contain a lot of local minimums and can confound a gradient in directions of growth, natural gradient descent is suitable. Moreover, the accuracy is more qualified. It happens by taking into account the gradient directions and the curvature properties of the optimizing function, which allows avoiding local minimums. Hence, this optimization algorithm can be applied in neural networks, where the advantage of natural gradient descent over Adam can improve and accelerate the learning process.

In further research, we can examine the behavior of Algorithm 3 with the Fisher matrix of other distributions, such as gamma, Gompertz, or Gumbel distributions. Moreover, we can reduce the Riemannian gradient flow to another smooth manifold other than the manifold of the probability distribution, which can potentially facilitate the optimization method.

Author Contributions: Conceptualization, R.A.; Formal analysis, R.A.; Funding acquisition, P.L., N.N.;Investigation, R.A.; Methodology, P.L.; Project administration, P.L.,

N.N; Resources, R.A.; Supervision, P.L.Writing—original draft, R.A.; Writing—review editing, P.L. All authors have read and agreed to the published version of the manuscript.

Funding: The research in section 3 was supported by the Russian Science Foundation (Project No. 21-71-00017). The research in section 4 was supported by the Russian Science Foundation (Project No. 22-71-00009). The remaining of the work is supported by North-Caucasus Center for Mathematical Research with the Ministry of Science and Higher Education of the Russian Federation.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgment: The authors would like to thank the North-Caucasus Federal University for supporting in the contest of projects competition of scientific groups and individual scientists of the North-Caucasus Federal University.

Conflicts of Interest: The authors declare no conflict of interest.

1. Ward, R.; Wu, X.; Bottou, L. AdaGrad Stepsizes: Sharp Convergence Over Nonconvex Landscapes. *Journal of Machine Learning Research 21* **2020**, *21*, 1–30.
2. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* **2011**, *12*, 2121—2159.
3. Xu, D.; Zhang, S.; Zhang, H.; Mandic, D.P. Convergence of the RMSProp deep learning method with penalty for nonconvex optimization. *Neural Networks* **2021**, *139*, 17–23.
4. Qu, Z.; Yuan, S.; Chi, R.;Chang, L.; Zhao, L. Genetic Optimization Method of Pantograph and Catenary Comprehensive Monitor Status Prediction Model Based on Adadelta Deep Neural Network. *IEEE access* **2019**, *7*, 23210–23221.
5. Wu, H.-P.; Li, L. The BP Neural Network with Adam Optimizer for Predicting Audit Opinions of Listed Companies, *IAENG International Journal of Computer Science* **2021**, *48*, 364–368.
6. Toussaint, M. Some notes on gradient descent. Lecture Notes, 2012.
7. Wang, S.; Teng, Y.; Perdikaris, P. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing* **2021**, *43*, 3055—3081.
8. Martens, J. New Insights and Perspectives on the Natural Gradient Method. *Journal of Machine Learning Research* **2020**, *21*, 1–76.
9. Huang, Y.; Zhang, Y.; Chambers, J.A. A Novel Kullback–Leibler Divergence Minimization-Based Adaptive Student's t-Filter. *IEEE Transactions on Signal Processing* **2019**, *67*, 5417–5432.
10. Asperti, A.; Trentin, M. Balancing Reconstruction Error and Kullback-Leibler Divergence in Variational Autoencoders. *IEEE Access* **2019**, *8*, 199440–199448.
11. Heck, D. W.; Moshagen, M.; Erdfelder, E. Model selection by minimum description length: Lower-bound sample sizes for the Fisher information approximation. *Journal of Mathematical Psychology* **2014**, *60*, 29–34.
12. Spall, J.C.Monte Carlo Computation of the Fisher Information Matrix in Nonstandard Settings. *Journal of Computational and Graphical Statistics* **2012**, *14*, 889–909.
13. Alvarez, F.; Bolte, J.; Brahic, O. Hessian Riemannian Gradient Flows in Convex Programming. *Society for Industrial and Applied Mathematics* bf 2004, *43*, 68–73.
14. Bah, B.; Rauhut, H; Terstiege, U.; Westdickenberg, M. Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers graphic. *Information and Inference: A Journal of the IMA* **2021**, *11*, 307–353.
15. Alsuroji, R.; Bouguila, N.; Zamzami, N. Predicting Defect-Prone Software Modules Using Shifted-Scaled Dirichlet Distribution. *2018 First International Conference on Artificial Intelligence for Industries (AI4I)* **2018**, 15–18.
16. Wong, T.-T. Generalized Dirichlet distribution in Bayesian analysis. *Applied Mathematics and Computation* **1998**, *97*, 165–181.