

Решение уравнения Пуассона с помощью физико-информированной нейронной сети с натуральным градиентным спуском с распределением Дирихле

Р.И. Абдулкадиров, П.А. Ляхов, Н.Н. Нагорнов

Северо-Кавказский федеральный университет, Ставрополь

Аннотация: В данной статье предлагается физико-информированная нейронная сеть, содержащая натуральный градиентный спуск, для решения краевой задачи уравнения Пуассона. Методы машинного обучения, использующийся в решении дифференциальных уравнений в частных производных, являются альтернативой по отношению методу конечных элементов. Традиционные численные методы решения дифференциальных уравнений не способны с эквивалентной эффективностью решать произвольные задачи математической физики, в отличие от методов машинного обучения. За точность решения начальных и краевых задач уравнений в частных производных отвечает функция потерь нейронной сети. Чем эффективнее проходит минимизация функции потерь, тем более точное решение получается в итоге. Самым традиционным алгоритмом оптимизации является адаптивная оценка моментов, которая по сей день используется в глубоком обучении. Однако данный подход не гарантирует достижения глобального минимума функции потерь. Вследствие этого, мы предлагаем использовать натуральный градиентный спуск с распределением Дирихле, который позволил повысить точность решения уравнения Пуассона.

Ключевые слова: натуральный градиентный спуск, уравнение Пуассона, матрица Фишера, метод конечных элементов, нейронные сети.

Введение

Проблема поиска точных или численных решений дифференциальных и интегральных уравнений остается актуальной. Существует множество аналитических методов решения различных уравнений в частных производных (УЧП), отмеченных в [1 – 3]. К точным методам относятся классические и неклассические редукции симметрии, дифференциальные ограничения, разделение переменных и подходы Пенлеве. Это облегчает изучение УЧП, но, к сожалению, не всегда удается построить аналитическое решение. Тогда приходится применять численные методы.

На практике многие исследователи применяют методы конечных разностей, конечных элементов (МКЭ) и конечных объемов для решения начальных и граничных условий УЧП в [2]. Первый метод является самым

ранним среди других и требует много времени для вычислений. Второй и третий методы эффективны для моделирования, существенно сокращая время вычислений. Более того, МКЭ является наиболее предпочтительным подходом для УЧП в 2 и 3 измерениях.

Новейшим подходом поиска решений УЧП является физико-информированная нейронная сеть (PINN) [4]. Она состоит из обычного многослойного персептрона и слоев автоматической дифференциации. В соответствии со значениями, полученными от УЧП и его начального и граничного условия, PINN корректирует решение на каждой итерации.

В данной статье мы предлагаем PINN, содержащую натуральный градиентный спуск (NGDM Nesterov) с распределением Дирихле, для повышения точности решения уравнения Пуассона и сокращения времени обучения по сравнению с МКЭ. В экспериментальной части демонстрируется преимущество предложенного NGDM Nesterov над адаптивной оценкой моментов (Adam). Кроме того, мы показываем, что PINN с NGDM Nesterov достигает точности выше, чем МКЭ.

Предварительные сведения

Мы рассматриваем следующее УЧП с параметром для решения $u(x)$ с $x \in \Omega \subset R^n$:

$$f\left(x; \frac{\partial u}{\partial x_1}, \dots, \frac{\partial u}{\partial x_n}; \frac{\partial^2 u}{\partial x_1 \partial x_1}, \dots, \frac{\partial^2 u}{\partial x_1 \partial x_n}; \dots; \lambda\right) = 0, x \in \Omega,$$

с граничным условием:

$$B(u, x) = 0, x \in \partial\Omega,$$

где $B(u, x)$ может быть условием Дирихле, Неймана или Робина.

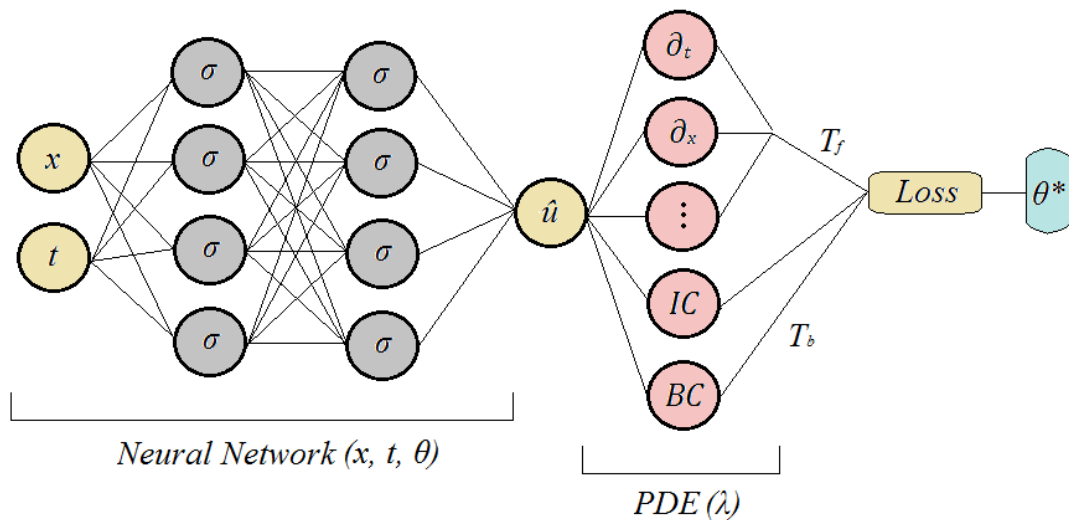


Рис. 1. – Архитектура физико-информированной сети

На рисунке 1 представлена архитектура PINN. Эта конструкция принимает на входных слоях значения пространственных и временных переменных x и t , соответственно. Далее эти значения проходят через скрытые слои, а затем на последнем нейроне нейронная сеть выдает значение \hat{u} . Затем выходные результаты нейронной сети поступают на слой, который состоит из частных производных произвольных порядков и начальных и граничных условий (IC и BC). Обратите внимание, что $\theta = \{W^{(l)}, b^{(l)}\}_{1 \leq l \leq k}$ – это набор всех весовых матриц $W^{(l)}$ и векторов смещения $b^{(l)}$ в нейронной сети, а также параметры θ , полученные путем минимизации функции потерь $L(\theta; T)$. Необходимо ограничить нейронную сеть, налагаемой УЧП и граничными условиями. На практике мы ограничиваемся некоторыми разбросанными точками (например, случайно распределенными точками или кластеризованными точками в области определения), то есть обучающими данными размером $T = \{x_1, \dots, x_T\}$. Кроме того, состоит из двух множеств $T_f \in \Omega$ и $T_b \in \partial\Omega$, которые являются точками в области и на границе соответственно. Мы будем называть T_f и T_b множествами «остаточных точек».

Чтобы измерить несоответствие между нейронной сетью и ограничениями, мы рассматриваем функцию потерь, определяемую, как взвешенное суммирование L_2 норм для УЧП и граничных условий:

$$L(\theta; T) = w_f L_f(\theta; T_f) + w_b L_b(\theta; T_b)$$

где

$$L_f(\theta; T_f) = \sum_{x \in T_f} \left\| f \left(x; \frac{\partial \mathcal{L}}{\partial x_1}, \dots, \frac{\partial \mathcal{L}}{\partial x_n}; \frac{\partial^2 \mathcal{L}}{\partial x_1 \partial x_1}, \dots, \frac{\partial^2 \mathcal{L}}{\partial x_1 \partial x_n}; \dots; \lambda \right) \right\|_2^2$$

и

$$L_b(\theta; T_b) = \sum_{x \in T_b} \|B(\mathcal{L}, x)\|_2^2,$$

где w_f и w_b — веса УЧП и граничных условий. Для сохранения физического

смысла УЧП, функция потерь содержит производные, такие как $\frac{\partial \mathcal{L}}{\partial x_i}$ и

$\frac{\partial^2 \mathcal{L}}{\partial x_i \partial x_j}$, которые обрабатываются посредством автоматического дифференцирования.

Нейронные сети, решающие дифференциальные и интегральные уравнения в частных производных, можно модифицировать, выбрав новый оптимизатор, расширив функции потерь и преобразовав уровни в сети. На практике сеть на рисунке 1 можно заменить на рекуррентные, модульные, вейвлетные или комплекснозначные нейронные сети, но этот подход подходит в случае уравнений Бюргерса, Клейна-Гордона, КдВ и Навье-Стокса.

Алгоритм Adam (Adaptive Moment Estimation, адаптивная оценка момента) из [5] вычисляет индивидуальные скорости адаптивного обучения для различных параметров на основе оценок из первого и второго моментов градиентов. Данный алгоритм является одним из наиболее широко применяемых в сверточных, остаточных и многих других нейронных сетях.

Также он имеет применение в PINN, что было продемонстрировано в [6]. В Алгоритме 1 мы представляем псевдокод Adam.

Алгоритм 1 Адаптивная оценка момента

Входные данные: γ (скорость обучения), β_1, β_2 (моменты первого и второго порядка), θ_0 (начальное значение), f (функция потерь), λ (распад веса)

Выходные данные: θ_n (результат)

1. $m_0 \leftarrow 0$ (первый момент), $v_0 \leftarrow 0$ (второй момент)
 2. **for** i from 1 to n **do**
 3. $g_i \leftarrow \nabla f(\theta_{i-1}) + \lambda \theta_{i-1}$
 4. $m_i \leftarrow \beta_1 m_{i-1} + (1 - \beta_1) g_i$
 5. $v_i \leftarrow \beta_2 v_{i-1} + (1 - \beta_2) g_i^2$
 6. $\hat{m}_i \leftarrow \frac{m_i}{(1 - \beta_1^i)}$
 7. $\hat{v}_i \leftarrow \frac{v_i}{(1 - \beta_2^i)}$
 8. $\theta_i \leftarrow \theta_{i-1} - \gamma \frac{\hat{m}_i}{(\sqrt{\hat{v}_i} + \epsilon)}$
 9. **end for**
-

Отметим, что существуют две модификации адаптивной оценки моментов, такие как NAdam [7] и RAdam [8]. Они лучше работают в сверточных нейронных сетях, но в PINN такие алгоритмы не эффективны. В [6] лучшие результаты были получены благодаря Adam, а не NAdam или RAdam. Но недостатком Adam в PINN являются слишком большие затраты времени, что не позволяет конкурировать с МКЭ. Но если уменьшить оценку моментов и учесть не только направления градиентов, но и кривизну функции потерь, то можно уменьшить затраты времени и существенно повысить точность решения.

Натуральный градиентный спуск с распределением Дирихле

Алгоритм натурального градиентного спуска, способный ускорить процесс обучения нейронных сетей, рассмотрен в [9]. Напомним итерационную формулу с размером шага γ_k :

$$\theta_{k+1} = \theta_k - \gamma_k F(\theta_k)^{-1} \nabla f(\theta_k).$$

Основной частью естественного градиентного спуска является матрица Фишера $F(\theta_k)$ из [10], которую можно вычислить на многообразии вероятностных распределений с помощью дивергенции Кульбака-Лейблера для любой функции $f(\theta_k)$. Пусть $p(x; \xi)$ – некоторое семейство вероятностных распределений по x , параметризованное вектором действительных чисел $\xi_t + \delta\xi$. Тогда мы получаем расхождение $K-L$ для непрерывного распределения вероятностей:

$$KL(p(x; \xi_t); p(x; \xi_t + \delta\xi)) = -\frac{1}{2} \delta\xi^T E[\nabla \log p(x; \xi_t) \nabla \log p(x; \xi_t)^T] \delta\xi$$

где $E[\nabla \log p(x; \xi_t) \nabla \log p(x; \xi_t)^T] = F(\xi_t)$ – информационная матрица Фишера, которая представляет собой Риманову структуру на многообразии вероятностных распределений [11] и [12].

Распределение Дирихле порядка $K \geq 2$ с параметрами $\alpha_1, \dots, \alpha_k > 0$ [13] имеет функцию плотности вероятности относительно меры Лебега на Евклидовом пространстве R^{K-1} , определяемую выражением:

$$f(x; a) = \frac{1}{B(a)} \prod_{i=1}^K x_i^{\alpha_i - 1}, B(a) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^K \alpha_i)},$$

где $\{x_i\}_{i=1}^K$ принадлежит $(K-1)$ -симплексу.

Следовательно, мы можем найти матрицу Фишера F из [14] с $F_{ii} = \psi'(\alpha_i) - \psi'(\sum_{j=1}^K \alpha_j)$ и $F_{il} = -\psi'(\sum_{j=1}^K \alpha_j)$, где $\psi' = \frac{d^2}{dx^2} \log(\Gamma(x))$ – тригамма-функция и $i, l = 1, \dots, K, i \neq l$.

Согласно информационной матрице Фишера распределения Дирихле получаем Алгоритм 2.

Алгоритм 2

Натуральный градиентный спуск с моментом и распределением Дирихле

Входные данные: γ (скорость обучения), θ_0 (начальное значение), f (функция потерь), λ (распад веса), μ (импульс), τ (демпфирование), F (матрица Фишера).

Выходные данные: θ_n (результат)

1. **for** i from 1 to n **do**
 2. $g_i \leftarrow \nabla f(\theta_{i-1}) + \lambda \theta_{i-1}$
 3. **if** $i > 1$ **then**
 4. $b_i \leftarrow \mu b_{i-1} + (1 + \tau) g_i$
 5. **else**
 6. $b_i \leftarrow g_i$
 7. **end if**
 8. $\theta_i \leftarrow \theta_{i-1} - \gamma F^{-1}(g_{i-1} + \mu b_i)$
 9. **end for**
-

Заметим, что в Алгоритме 2 нет необходимости уменьшать длину шагов или числовое значение градиента для улучшения конечных значений экстремумов. Матрица Фишера содержит параметры без элементов вектора x , что позволяет избежать дополнительных вычислений в цикле. Учитывая свойства кривизны, натуральные градиенты достигают глобального минимума быстрее, что позволяет за короткое время добиться более высокой точности решения в PINN.

Эксперимент

В экспериментальной части демонстрируются точности решений, полученных МКЭ, PINN(Adam) и PINN (NGDM Nesterov).

Рассмотрим уравнение Пуассона:

$$-\Delta u(x, y) = -1, \Omega = [0,1]^2,$$

$$u(x, y) = 0, \partial\Omega.$$

Точное решение краевой задачи, которое можно получить методом разделения, имеет вид:

$$u(x, y) = \int_0^1 \int_0^1 G(x, y, \xi, \eta) d\eta d\xi,$$

$$G(x, y, \xi, \eta) = \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \frac{\sin(p_n x) \sin(q_m y) \sin(p_n \xi) \sin(q_m \eta)}{p_n^2 + q_m^2},$$

где $p_n = \pi n$, $q_m = \pi m$.

Решение уравнения Пуассона, заданное PINN, представлено на рис. 2. Теперь сравним решения, полученные МКЭ и PINN, содержащие Adam и NGDM Nesterov (таблица 1).

Таблица № 1

Сравнение решений, представленных МКЭ и PINN (Adam и NGDM)

Метод решения	Ошибка	Сетка, количество эпох	Время, с
FEM	1,378e-4	квадрат 256×256	61,1
PINN (Adam)	2,098e-4	10000	95,3
PINN (NGDM Nesterov)	8,911e-5	10000	66,7

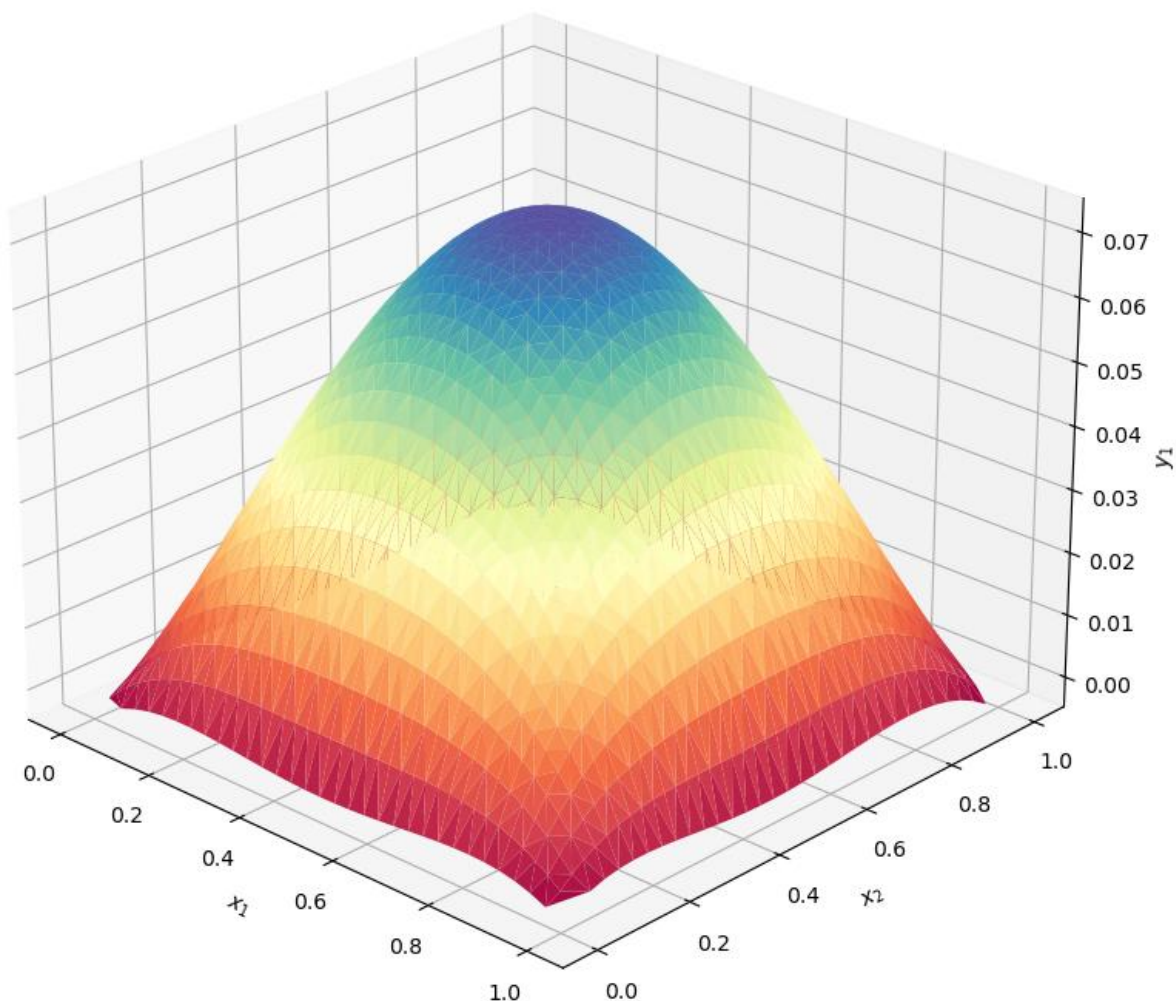


Рис. 2. – Решение уравнения Пуассона, заданное PINN

В таблице 1 показано, что PINN (NGDM) достигает максимальной точности за 66,7 секунды, что довольно быстро для уравнения Пуассона. Но он решает уравнение медленнее, чем МКЭ, который использует сетку из квадратов 256×256 и обеспечивает хорошую точность. Если мы возьмем сетку с квадратами больше 256, это займет слишком много времени. PINN (Adam) решает первую краевую задачу за 95,3 секунды.

Заключение

По результатам экспериментов можно сделать вывод, что PINN (NGDM Nesterov) дает наибольшую точность решения за короткое время. Это означает, что предлагаемый подход может конкурировать с МКЭ по

точности и времени. В ходе дальнейших исследований мы сможем рассчитать информационные матрицы Фишера распределений Гумбеля и Гомпертца, которые потенциально могут быть более эффективными, чем распределение Дирихле. Более того, вместо PINN мы будем применять глубокую операторскую сеть (DeepONet) для решения нелинейных уравнений в частных уравнениях, таких, как уравнения Клейна-Гордона, Бюргерса и КдВ.

Благодарности

Исследование проведено при поддержке Совета по грантам Президента Российской Федерации (проект № МК-371.2022.4).

Литература

1. Полянин А.Д., Журов А.И. Методы разделения переменных и точные решения нелинейных уравнений математической физики. Москва. ИПМех РАН. 2020. 384 с.
2. Полянин А.Д. Справочник по линейным уравнениям математической физики. Москва. Физ.-мат. лит., Наука/Интерпериодика. 2001. 575 с.
3. Полянин А.Д., Зайцев В.Ф. Справочник по нелинейным уравнениям математической физики. Москва. ФИЗМАТЛИТ. 2008. 688 с.
4. Raissia M., Perdikaris P., Karniadakis G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*. 2019. V. 378. Pp. 686-707.
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. V. 48. Pp. 364-368.



6. Lu L., Meng X., Mao Z., Karniadakis G.E. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*. V. 63. №1. Pp. 208-228.
7. Dozat T. Incorporating Nesterov Momentum into Adam. *Proceedings of the 4th International Conference on Learning Representations California*. 2016. Pp. 1-4.
8. Liu L., Jiang H., He P., Chen W., Liu X., Gao J., Han J. On the Variance of the Adaptive Learning Rate and Beyond. *International Conference on Learning Representations California*. 2020. Pp. 1-13.
9. Martens J. New Insights and Perspectives on the Natural Gradient Method. *Journal of Machine Learning Research*. 2020. V. 21. Pp. 1-76.
10. 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. V. 67. Pp. 5417-5432.
11. Alvarez F., Bolte J., Brahic O. Hessian Riemannian Gradient Flows in Convex Programming. *SIAM Journal on Control and Optimization*. 2004. V. 43. №2. Pp. 68-73.
12. Abdulkadirov R., Lyakhov P., Nagornov N. Accelerating Extreme Search of Multidimensional Functions Based on Natural Gradient Descent with Dirichlet Distributions. *Mathematics*. 2022. V. 10. №19. P. 3556.
13. Wong T.T. Generalized Dirichlet distribution in Bayesian analysis. *Applied Mathematics and Computation*. 1998. V. 87. Pp. 165-181.
14. Abdulkadirov R.I., Lyakhov P.A. Improving Extreme Search with Natural Gradient Descent Using Dirichlet Distribution. *Mathematics and its Applications in New Computer Systems. Lecture Notes in Networks and Systems*. 2021. V. 424. Pp. 19-28.

References

1. Poljanin A.D., Zhurov A.I. Metody razdelenija peremennyh i tochnyt reshenija nelinejnyh uravnenij matematicheskoy fiziki [Separation of Variables and Exact Solutions to Nonlinear PDEs]. Moskva: IPMeh RAN, 2020. 384 p.
2. Poljanin A.D. Spravochnik po linejnym uravnenijam matematicheskoy fiziki [Handbook of Linear Partial Differential Equations for Engineers and Scientists]. Moskva: Fiz.-mat. lit., Nauka/Interperiodika, 2001. 575 p.
3. Poljanin A.D., Zajcev V.F. Spravochnik po nelinejnym uravnenijami matematicheskoy fiziki [Handbook of Nonlinear Partial Differential Equations]. Moskva: FIZMALIT, 2008. 688 p.
4. Raissia M., Perdikaris P., Karniadakis G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*. 2019. V. 378. Pp. 686-707.
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. V. 48. Pp. 364-368.
6. Lu L., Meng X., Mao Z., Karniadakis G.E. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*. V. 63. №1. Pp. 208-228.
7. Dozat T. Incorporating Nesterov Momentum into Adam. *Proceedings of the 4th International Conference on Learning Representations California*. 2016. Pp. 1-4.
8. Liu L., Jiang H., He P., Chen W., Liu X., Gao J., Han J. On the Variance of the Adaptive Learning Rate and Beyond. *International Conference on Learning Representations California*. 2020. Pp. 1-13.
9. Martens J. New Insights and Perspectives on the Natural Gradient Method. *Journal of Machine Learning Research*. 2020. V. 21. Pp. 1-76.



10. 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. V. 67. Pp. 5417-5432.

11. Alvarez F., Bolte J., Brahic O. Hessian Riemannian Gradient Flows in Convex Programming. *SIAM Journal on Control and Optimization*. 2004. V. 43. №2. Pp. 68-73.

12. Abdulkadirov R., Lyakhov P., Nagornov N. Accelerating Extreme Search of Multidimensional Functions Based on Natural Gradient Descent with Dirichlet Distributions. *Mathematics*. 2022. V. 10. №19. P. 3556.

13. Wong T.T. Generalized Dirichlet distribution in Bayesian analysis. *Applied Mathematics and Computation*. 1998. V. 87. Pp. 165-181.

14. Abdulkadirov R.I., Lyakhov P.A. Improving Extreme Search with Natural Gradient Descent Using Dirichlet Distribution. *Mathematics and its Applications in New Computer Systems. Lecture Notes in Networks and Systems*. 2021. V. 424. Pp. 19-28.