# Новый подход к обучению нейронных сетей с помощью натурального градиентного спуска с импульсом на основе распределений Дирихле

*Р.И. Абдулкадиров [1], П.А. Ляхов [2]*
*[1] Северо-Кавказский центр математических исследований, 355009, Россия, г. Ставрополь, ул. Пушкина 1;*
*[2] Северо-Кавказский федеральный университет, 355009, Россия, г. Ставрополь, ул. Пушкина 1*

***Аннотация***

В данной работе мы предлагаем алгоритм натурального градиентного спуска с импульсом на основе распределений Дирихле для ускорения обучения нейронных сетей. Данный подход учитывает не только направления градиентов, но и выпуклость минимизируемой функции, что значительно ускоряет процесс поиска экстремумов. Представлены вычисления натуральных градиентов, базирующихся на распределениях Дирихле, и реализовано внедрение предложенного подхода в схему обратного распространения ошибок. Результаты по распознаванию изображений и прогнозированию временных рядов во время проведения экспериментов показывают, что предложенный подход дает более высокую точность и не требует большого количества итераций для минимизации функций потерь, по сравнению с методами стохастического градиентного спуска, адаптивной оценки момента и адаптивным по параметрам диагональным квазиньютоновским методом для невыпуклой стохастической оптимизации.

*Ключевые слова*: распознавание образов, машинное обучение, оптимизация, распределения Дирихле, натуральный градиентный спуск.

*Цитирование*: **Абдулкадиров, Р.И**. Новый подход к обучению нейронных сетей с помощью натурального градиентного спуска с импульсом на основе распределений Дирихле / Р.И. Абдулкадиров, П.А. Ляхов // Компьютерная оптика. – 2023. – Т. 47, № 1. – С. 160-169. – DOI: 10.18287/2412-6179-CO-1147.

*Citation*: Abdulkadirov RI, Lyakhov PA. A new approach to training neural networks using natural gradient descent with momentum based on Dirichlet distributions. Computer Optics 2023; 47(1): 160-169. DOI: 10.18287/2412-6179-CO-1147.

## *Введение*

Наиболее важную роль в искусственных нейронных сетях играют методы оптимизации, которые существенно влияют на процесс обучения. Конечная точность в процессе обучения зависит от согласования значений весов искусственных нейронов с функцией потерь, которую с каждой эпохой необходимо минимизировать. Если оптимизация проходит быстро и сходится к глобальному минимуму, то повышается точность распознавания и сокращается время обучения.

Одним из самых известных методов оптимизации является стохастический градиентный спуск SGD [1], который был модифицирован в SGDM [2] и SGDM с условием Нестерова [3]. Позднее на базе градиентного подхода были предложены новые методы оптимизации: AdaGrad [4], ADADELTA [5], RMSProp[6] и Adam [7]. В настоящее время наиболее распространенные методы в машинном обучении – это SGDM с модификацией Нестерова и Adam.

Достижение глобального минимума за меньшее количество итераций (эпох) с требуемой точностью по сей день остается актуальной проблемой в методах оптимизации. Особенно остро встает вопрос нахождения минимума в машинном обучении, где процесс оптимизации функции потерь влияет на конечную точность. Для решения данной проблемы был предложен градиентный поток из [8], представляющий собой произведение метрического тензора на гладком многообразии и градиента оптимизируемой функции. Такой подход ускорил процесс минимизации функции потерь в нейронных сетях, но в данной статье будут использоваться многообразия вероятностных распределений вместо гладких.

Многообразия вероятностных распределений в основном используются в информационной геометрии, где аналогом градиентного потока является натуральный градиент. Натуральный градиент представляет собой произведение информационной матрицы Фишера и градиента оптимизируемой функции. Матрица Фишера рассчитывается по расхождению Кульбака–Лейблера (расхождение K-L в [9] и [10]).

Натуральный градиентный спуск с импульсом (NGDM) является альтернативой стохастическому градиентному спуску и его модификациям, как было отмечено в [11]. Благодаря натуральному градиенту, содержащему матрицу Фишера, базирующуюся на вероятностном распределении, процесс оптимизации сходится в области глобального минимума с высокой точностью. Вероятностные распределения стоит выбирать таким образом, чтобы матрица Фишера содержала только постоянные значения. В данной ста-

тье мы предлагаем алгоритм обучения нейронных сетей с помощью натурального градиентного спуска с импульсом на основе распределения Дирихле и обобщенного распределения Дирихле. Мы покажем, что предложенный подход имеет более высокую точность и не требует большого количества итераций для минимизации функций потерь, в отличие от SGDM, Adam и Apollo [12]. Затем продемонстрируем работу предложенного алгоритма в экспериментах с распознаванием образов и прогнозированием временных рядов. В заключении мы обсудим результаты, перспективы и направления разработок новых модификаций натурального градиентного спуска.

### *1. Предварительные сведения*

Пусть $f: \Omega \to \mathbb{R}$ – гладкая функция над замкнутым выпуклым множеством $\Omega \in \mathbb{R}^n$, содержащая один или несколько экстремумов. Задача стохастического градиентного спуска состоит в нахождении наименьшего значения функции $f(\theta)$ в заданной области $\Omega$ с помощью следующей итеративной формулы:

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k \nabla f\left(\theta^{(k)}\right), \tag{1}$$

где $\theta$ – произвольный аргумент.

Функция $f$ может быть минимизирована с помощью SGDM с модификацией Нестерова из [3], псевдокод которого представлен в Алгоритме 1.

**Algorithm 1.** Стохастический градиентный спуск с импульсом и модификацией Нестерова

**Input:** $\gamma$ (скорость обучения), $\theta_0$ (входные параметры), $f$ (целевая функция), $\lambda$ (распад веса), $\mu$ (импульс), $\tau$ (демпфирование)

**Output:** $\theta_n$ (конечный результат)

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

Стохастический градиентный спуск с импульсом и модификацией Нестерова очень практичен в сверточных нейронных сетях для распознавания изображений. Процесс минимизации при SGDM не требует много времени и ресурсов, но достичь глобального минимума у него не всегда удается.

Наиболее предпочтительным методом оптимизации в нейронных сетях, решающих не только задачи распознавания образов, является Adam. Данный метод отличается от SGDM и является более надежным в машинном обучении, потому что он обновляет экспоненциальные скользящие средние градиента $m_t$ и квадрата градиента $\upsilon_t$ с гиперпараметрами $\beta_1, \beta_2 \in [0, 1)$, контролирующими скорость экспоненциального затухания этих скользящих средних. Однако эти скользящие средние инициализируются как (векторы) нули, что приводит к оценкам моментов, которые смещены в сторону нуля, особенно на начальных шагах и при малых скоростях затухания. Псевдокод метода Adam ([7]) представлен в Алгоритме 2.

**Algorithm 2.** Адаптивная оценка момента (Adam)

**Input:** $\gamma$ (скорость обучения), $\beta_1$, $\beta_2$ (коэффициенты для вычисления скользящих средних градиента и его квадрата), $\theta_0$ (входные параметры), $f$ (целевая функция), $\lambda$ (распад веса)

**Output:** $\theta_n$ (конечный результат)

1: $m_0 \leftarrow 0$ (первый момент), $\upsilon_0 \leftarrow 0$ (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: $\hat{m}_i \leftarrow m_i / \left(1-\beta_1^i\right)$
7: $\hat{v}_i \leftarrow v_i / \left(1-\beta_2^i\right)$
8: $\theta_i \leftarrow \theta_{i-1} - \gamma\hat{m}_i / \left(\sqrt{\hat{v}_i} + \epsilon\right)$
9: **end for**

Метод Adam широко используется в библиотеках машинного обучения MATLAB, Python и R, но и он не лишен недостатков. Для спуска в область глобального минимума требуется много итераций, а иногда глобальный экстремум не достигается вообще. Помимо Adam, в задачах распознавания изображений может использоваться алгоритм Apollo [12]. Данный подход отличается от представленных выше тем, что путем аппроксимации матрицы Гесса он способен учитывать выпуклость минимизируемой функции. Основное преимущество Apollo состоит в том, что он способен уменьшать стохастическую дисперсию, что упрощает аппроксимацию матрицы Гесса, сохраняет положительную определенность в условии невыпуклости целевой функции и сходится в выпуклой и стохастической оптимизациях.

**Algorithm 3.** Apollo Адаптивный по параметрам диагональный квазиньютоновский метод невыпуклой стохастической оптимизации

**Input:** $\gamma$ (скорость обучения), $\beta$ (коэффициенты для вычисления скользящего среднего градиента), $\theta_0$ (входные параметры), $f$ (целевая функция), $\epsilon = 10^{-4}$

**Output:** $\theta_n$ (конечный результат)

1: $m_0 \leftarrow 0$ (скользящее среднее с поправкой на смещение), $d_0 \leftarrow 0$ (коррекция направления), $B_0 \leftarrow 0$ (аппроксимация Гессиана)
2: **for** $i$ from 1 to $n$ **do**

3: $g_{i+1} \leftarrow \nabla f(\theta_i)$

4: $m_{i+1} \leftarrow \frac{\beta(1-\beta^i)}{1-\beta^{i+1}} m_i + \frac{1-\beta}{1-\beta^{i+1}} g_{i+1}$

5: $\alpha \leftarrow \frac{d_i^T (m_{i+1} - m_i) + d_i^T B_i d_i}{(\| d_i \|_4 + \epsilon)^4}$

// значения коэффициента для $B$

6: $B_{i+1} \leftarrow B_i - \alpha \cdot Diag(d_i^2)$

7: $D_{i+1} \leftarrow rectify(B_{i+1}, 0.01)$ // устранение невыпулости

8: $d_{i+1} \leftarrow D_{i+1}^{-1} m_{i+1}$

9: $\theta_{i+1} \leftarrow \theta_i - \gamma d_{i+1}$

0: **end for**

В Алгоритмах 1 и 2 направление к минимуму определяется с помощью градиентов. Но если, как в Алгоритме 3, учитывать не только поле градиентов, но и выпуклость поверхности, описанной оптимизируемой функцией *f*, то это даст возможность достигать именно глобального минимума с требуемой точностью. Но данный подход будет аппроксимировать Гессиан каждую итерацию, что увеличит количество вычислений и временные затраты. Далее мы изложили наш подход к решению этой проблемы, используя натуральный градиент.

## *2. Метод быстрого поиска экстремума на основе NGDM и распределений Дирихле*

### *2.1. K-L расхождение для NGDM*

Натуральный градиентный спуск ([11], [13]) с импульсом, удовлетворяющий условию Нестерова, может быть представлен следующим образом:

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k F^{-1} \left( \nabla f \left( \theta^{(k)} \right) + \mu b^{(k+1)} \right), \tag{2}$$

где $\theta^{(0)} = \theta_0$ – начальная точка, $b^{(k+1)} = \mu b^{(k)} + (1-\tau)(\nabla f(\theta^{(k)}) + \lambda \theta^{(k)})$ ($\tau$ – параметр демпфирования), $F$ – матрица Фишера, которая учитывает кривизну поверхности $f$ для обхода локальных минимумов и отличает натуральный градиентный спуск (2) от стохастического (1). Определение матрицы Фишера берет свое начало еще с определения градиентного потока на гладких Римановых многообразиях в [8], где свойства производных (градиентов) и кривизны уже рассмотрены в общих случаях. Данный подход уже пытались использовать в методах оптимизации в [14]. Впоследствии выяснилось, что наиболее эффективно оказалось использовать многообразия вероятностных распределений, где градиентным потоком является информационная матрица Фишера, вычисление которой можно провести с помощью расхождения Кульбака–Лейблера (K-L-расхождение).

Предположим, что $p(x;\xi)$ – некоторое семейство вероятностных распределений над значениями весовых коэффициентов $x$, где $\xi \in \mathbb{R}^n$ – вектор значений параметров распределения, регулирующих значения весовых коэффициентов. Тогда непрерывное K-L-расхождение имеет следующий вид [15]:

$$KL(p(x;\xi_t) \,\|\, p(x;\xi_t + \delta\xi)) = \frac{1}{2} \delta\xi^T F \delta\xi, \tag{3}$$

где $F = -\mathbb{E}[\nabla \log p(x;\xi_t) \; \nabla \log p(x;\xi_t)^T]$ – информационная матрица Фишера, представляющая собой градиентный поток на многообразии вероятностных распределений. Далее приведем расчеты матрицы Фишера для распределения Дирихле и обобщенного распределения Дирихле.

### *2.2. Вычисление матрицы Фишера с распределением Дирихле и его обобщением*

Распределение Дирихле порядка $K \geq 2$ с параметрами $\xi = \alpha$, где $\alpha_1, \ldots, \alpha_K > 0$ в [16] имеет функцию плотности вероятности относительно меры Лебега на Евклидовом пространстве $\mathbb{R}^{K-1}$, заданную формулой

$$p(x;\alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}, \; B(\alpha) = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma\left(\sum_i \alpha_i\right)}, \tag{4}$$

где $\{x_i\}_{i=1}^K$ удовлетворяет $\Sigma_i x_i = 1$, и $\Gamma(\alpha)$ – гамма-функция.

Вычислим логарифм от функции плотности (4).

$$\log p(x;\alpha) = \log \left[ \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_i \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \right] =$$

$$= \log \Gamma\left( \sum_{i=1}^{K} \alpha_i \right) - \sum_{i=1}^{K} \log \Gamma(\alpha_i) + \sum_{i=1}^{K} (\alpha_i - 1) \log x_i.$$

Вычислим частные производные второго порядка от $\log p(x;\alpha)$ по $\alpha$:

$$\frac{\partial^2}{\partial \alpha_j \partial \alpha_k} \log p = \psi' \left( \sum_{i=1}^{K} \alpha_i \right),$$

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

Следовательно, матрицу Фишера можно представить следующим образом:

$$F_{Dir}(\alpha) = \begin{pmatrix} \psi'(\alpha_1) - \psi'\left(\sum_i \alpha_i\right) & \ldots & -\psi'\left(\sum_i \alpha_i\right) \\ \ldots & \ldots & \ldots \\ -\psi'\left(\sum_i \alpha_i\right) & \ldots & \psi'(\alpha_K) - \psi'\left(\sum_i \alpha_i\right) \end{pmatrix}, \tag{5}$$

где $\psi(\alpha) = (d/d\alpha) \log(\Gamma(\alpha))$ – пси-функция.

Приведем обобщенное распределение Дирихле [16] для $\{x_i\}_{i=1}^K$, $\Sigma_i x_i = 1$, и $\alpha_i > 0$, $\beta_i > 0$, $i = 1, \ldots, K-1$, имеющее функцию плотности вероятности

$$p(x;\alpha,\beta) = \prod_{i=1}^{K} \frac{1}{B(\alpha_i, \beta_i)} x_i^{\alpha_i - 1} \left( 1 - \sum_{j=1}^{i} x_j \right)^{\gamma_i}, \tag{6}$$

где $\gamma_i = \beta_i - \alpha_{i+1} - \beta_{i+1}$ для $i = 1, ..., K-1$ и $\gamma_K = \beta_{K-1}$. Логарифм от функции плотности обобщенного распределения Дирихле имеет следующий вид:

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

Вычислим частные производные второго порядка от $\log p(x; \alpha, \beta)$:

$$1)\ \frac{\partial^2}{\partial\alpha_j\partial\alpha_l}\log p = \frac{\partial^2}{\partial\beta_j\partial\beta_l}\log p = \frac{\partial^2}{\partial\alpha_j\partial\beta_l}\log p = 0,\ j \neq l;$$
$$2)\ \frac{\partial^2}{\partial\alpha_j^2}\log p = \psi'(\alpha_j+\beta_j) - \psi'(\alpha_j);$$
$$3)\ \frac{\partial^2}{\partial\beta_j^2}\log p = \psi'(\alpha_j+\beta_j) - \psi'(\beta_j);$$
$$4)\ \frac{\partial^2}{\partial\alpha_j\partial\beta_j}\log p = \frac{\partial^2}{\partial\beta_j\partial\alpha_j}\log p = \psi'(\alpha_j+\beta_j).$$

Тогда матрица Фишера для обобщенного распределения Дирихле выражается следующим образом:

$$F_{GenDir}(\alpha) = \begin{pmatrix} \Psi_1 & ... & \mathcal{O} \\ ... & ... & ... \\ \mathcal{O} & ... & \Psi_K \end{pmatrix}, \qquad (7)$$

где

$$\Psi_i = \begin{pmatrix} \psi'(\alpha_i) - \psi'(\alpha_i+\beta_i) & -\psi'(\alpha_i+\beta_i) \\ -\psi'(\alpha_i+\beta_i) & \psi'(\beta_i) - \psi'(\alpha_i+\beta_i) \end{pmatrix}$$

и $\mathcal{O}$ – нулевая матрица.

Рассмотрим биективное отображение значений весовых коэффициентов $x$ на множество значений $z$, удовлетворяющих условию $\Sigma_j z_j = 1$. Пусть значения весовых коэффициентов $x$ принадлежат $n$-мерному шару $B(u_0, r)$, где $u_0 \in \mathbb{R}^n$, $r \in [0, \infty)$ – центр шара и его радиус соответственно. Из геометрии и топологии известно, что произвольный $n$-мерный шар $B(u_0, r)$ можно биективно отобразить на $n$-мерный симплекс $\Delta^n$:

$$B(u_0, r) \ni (x_1, ..., x_n) \mapsto (z_0, z_1, ..., z_n) \in \Delta^n.$$

После чего подбираем значения параметров $\alpha$ для распределения Дирихле, $\alpha$ и $\beta$ для обобщенного распределения Дирихле. При этом удовлетворяются условия двух критериев согласия [17]: энергетического критерия согласия, критерия согласия треугольника. Для значений $z$, удовлетворяющих условию

$$\sum\nolimits_{j=0}^{n} z_j = 1,$$

можно вычислить матрицу Фишера распределений Дирихле $p(z; \alpha)$ и $p(z; \alpha, \beta)$. В следствие чего вычисляются матрицы Фишера (5) и (7). Так как отображение между $x$ и $z$ биективное, то имеется возможность минимизировать функцию потерь $E(x(z))$ по значениям $z$. С помощью натурального градиентного спуска находится такое значение $z$, при котором функция потерь $E$ принимает наименьшее значение в точке $x(z)$.

Отсюда можно сделать вывод: натуральные градиенты на основе распределений Дирихле (4) и (6) могут использоваться на различных значениях весовых коэффициентов. То есть для произвольных переменных $x$ из распределений $p(x; \alpha)$ и $p(x; \alpha, \beta)$ возможно применение натурального градиентного спуска на основе распределений Дирихле.

Значения матрицы Фишера выбираются в зависимости от типа нейронной сети. В случае многослойного персептрона, где для стохастического и натурального градиентного спуска наиболее эффективная скорость обучения находится в промежутке $[0{,}01; 0{,}1]$, выбираются значения на отрезке от $[1; 4]$, так как в данной области матрица Фишера позволяет с большей точностью сходиться в области глобального минимума. Для сверточных и рекуррентных нейронных сетей, где наиболее эффективная скорость обучения находится в промежутке $[0{,}001; 0{,}005]$, выбираются значения на отрезке $[4; 8]$, что дает возможность «избегать» локальных минимумов.

Значения $\alpha_i$ на промежутке $[12; \infty)$ не способны регулировать значения весовых коэффициентов так же эффективно, как на отрезке $[4; 8]$. При значениях $\alpha_I = 12$ обратная матрица Фишера распределения Дирихле имеет следующий вид:

$$F_{Dir}^{-1} \approx diag\left(1/\psi'(12); ...; 1/\psi'(12)\right) \approx$$
$$\approx diag(1{,}428; ...; 1{,}428). \qquad (8)$$

Натуральный градиент с обратной матрицей Фишера (8) не способен минимизировать функцию потерь с требуемой точностью, так как увеличивается шаг. При $\alpha_i$, стремящейся к бесконечности, натуральный градиент становится «взрывающимся».

На промежутке $(0; 0{,}5]$ натуральный градиент принимает слишком малые значения, которые не позволяют сходиться в области глобального минимума. При $\alpha_i = 0{,}5$ обратная матрица Фишера имеет следующий вид:

$$F_{Dir}^{-1} \approx diag\left(1/\psi'(0{,}5); ...; 1/\psi'(0{,}5)\right) \approx$$
$$\approx diag(0{,}203; ...; 0{,}203). \qquad (9)$$

Натуральный градиент с обратной матрицей Фишера (9) не способен «обходить» локальные минимумы, так как значительно уменьшается шаг. При $\alpha_i$, стремящейся к 0, натуральный градиент становится «исчезающим». На отрезке $[0{,}5; 1]$ натуральный градиент не способен «обходить» ближайшие локальные минимумы, несмот-

ря на быструю сходимость. На отрезке [9; 12] натуральный градиент имеет возможность «обходить» локальные минимумы, но сходится с меньшей точностью.

Матрица Фишера $F_{GenDir}$ (7) является диагональной относительно блоков $\psi_i$. Вследствие чего наиболее эффективно выбирать значения $\alpha_i$ и $\beta_i$ на отрезке [3, 5; 9], где $\psi'(\alpha_i+\beta_i)$ приближенно равно 0. Следовательно,

$$F_{GenDir}^{-1} \approx$$
$$\approx diag\left(1/\psi'(\alpha_1);1/\psi'(\beta_1);...;1/\psi'(\alpha_n);1/\psi'(\beta_n)\right).$$

В случае многослойного персептрона выбираются значения на отрезке от [4, 5; 6], для возможности сходимости в области глобального минимума. Для сверточных и рекуррентных нейронных сетей выбираются значения на отрезке [5, 5; 9], что дает возможность «избегать» локальных минимумов.

После вычисления матриц Фишера для распределений Дирихле появляется возможность построить алгоритм натурального градиентного спуска с импульсом на основе распределений Дирихле, который будет внедрен в алгоритм обратного распространения ошибок.

### 2.3. Алгоритм поиска экстремума на основе NGDM и распределения Дирихле

В соответствии с матрицей Фишера для распределения Дирихле и обобщенного распределения Дирихле мы предлагаем Алгоритм 4 для ускоренного нахождения глобального минимума целевой функции $f$.

**Algorithm 4.** Натуральный градиентный спуск с импульсом, базирующийся на распределениях Дирихле.

**Input:** $\gamma$ (скорость обучения), $\theta_0$ (входные параметры), $f$ (целевая функция), $\lambda$ (распад веса), $\mu$ (импульс), $\tau$ (демпфирование), $F=F_{Dir}$ или $F=F_{GenDir}$ (матрица Фишера)

**Output:** $\theta_n$ (конечный результат)

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

Заметим, что в Алгоритме 4 нет необходимости уменьшать длину шага или числовое значение градиента для повышения точности за счет учета выпуклости минимизируемой функции. К тому же матрица Фишера содержит только параметры распределений без переменных $\theta$ и $x$ из формул (4) и (6), что позволяет избежать дополнительных вычислений в цикле и ресурсных затрат. Стоит отметить, что информационная матрица Фишера с обобщенным распределением Дирихле полезна только в случае $2n$-мерной поверхности, где $n\in\mathbb{N}$. Но для нейронных сетей такое ограничение не оказывает особого влияния.

### 3. Алгоритм обучения нейронной сети на основе NGDM и распределения Дирихле

В данном параграфе представлены алгоритм и схема предложенного метода обучения нейронной сети на основе натурального градиента. Из [18] вектор $x=\{x_1,...,x_m\}$, проходящий через нейрон $l$, приобретает значение вектора $y^{(l)}$. Затем сигнал $y^{(L)}$ сравнивается с ожидаемым выходом $d$. В итоге получаем результат ошибки $e_k=d_k-y_k$, где $k=1, ..., m$. Затем для достижения правильного ответа необходимо минимизировать функцию потерь $E(n)$, которая пошагово корректирует синаптические веса нейронов, пока система не достигнет устойчивого состояния. Псевдокод метода обратного распространения ошибок с использованием натурального градиентного спуска с импульсом представлен в векторной форме в Алгоритме 5.

Отметим, что в строке 2 Алгоритма 5 начинается прямое обучение, в 7 строке – обратное распространение ошибки, а с 14 по 19 строки содержится формула натурального градиентного спуска с импульсом, удовлетворяющая условию Нестерова. В 8 и 10 строках использована операция $\odot$ – умножение Адамара, которая выражается следующим образом:

$$(u_1,...,u_n)^T\odot(\upsilon_1,...,\upsilon_n)^T=(u_1\cdot\upsilon_1,...,\upsilon_n\cdot\upsilon_n)^T.$$

На рис. 1 продемонстрирована схема работы нейронной сети с обратным распространением ошибки, использующая для оптимизации натуральный градиентный спуск с импульсом. Благодаря информационной матрице Фишера, значения весов будут регулироваться лучше за счет учета не только направлений градиентов, но и выпуклости поверхности функции потерь $E(n)$.

Как видно на рис. 1, при обратном распространении ошибок веса нейронов будут принимать следующие значения:

$$w_k(n+1)=w_k(n)-\eta F^{-1}(\nabla_{w_k}E(n)+\mu b(n+1)), \quad (10)$$

где $b(n+1)=\mu b(n)+(1-\tau)(\nabla_{w_k}E(n)+\lambda w_k(n))$ ($\tau$ – параметр демпфирования), $\eta$ – скорость обучения, $F$ – матрица Фишера, $w_k\in\mathbb{R}^m$, $m=2, ...$ – вектор, выражающий веса. Аналогично можно применить метод обратного распространения ошибок для сверточных нейронных сетей, где будут регулироваться значения весов в сверточных, пулинговых и полносвязных слоях.

**Algorithm 5.** Алгоритм обратного распространения ошибок с NGDM Nesterov на основе распределений Дирихле

**Input:** $y_0 \in \mathbb{R}^m$ (входные данные), $d \in \mathbb{R}^m$ (вектор ожидаемых выходов), $w \in \mathbb{R}^m \times \mathbb{R}^m$ (весовые коэффициенты нейронов), $\phi$ (функция активации), $F$ (матрица Фишера), $\lambda$ (распад веса), $\mu$ (импульс), $\tau$ (демпфирование)

**Output:** $y^{(L)}$ (конечные результаты), $E(n)$ (функция потерь)

1: **for** $n$ from 1 to $N$ **do**
2: **for** $l$ from 1 to $L$ **do**
3: $y^{(l+1)}(n) \leftarrow \phi\, w^{(l)}(n) y^{(l)}(n)$
4: **end for**
5: $e(n) \leftarrow d(n) - y^{(L)}(n)$
6: **for** $l$ from 0 to $L$ **do**
7: **if** $l = L$ **then**

$$\delta^{(L)}(n) \leftarrow \frac{\partial E(n)}{\partial e(n)} \odot \frac{\partial y^{(L)}}{\partial v^{(L)}}$$

8:
9: else

$$\delta^{(l)}(n) \leftarrow w^{(l+1)}(n)\,\delta^{(l+1)}(n) \odot \frac{\partial y(n)}{\partial v(n)}$$

10:
11: **end if**
12: **for** $k$ from 0 to $m$ **do**
13: **if** $n > 1$ **then**
14: $b(n+1) \leftarrow \mu b(n) + (1-\tau) \times$
$\times(\delta^{(l)}(n) y^{(l-1)}(n) + \lambda w_k)$
15: **else**
16: $b(n+1) \leftarrow \delta^{(l)}(n) y^{(l-1)}(n+1) + \lambda w_k$
17: **end if**
18: $g^{(l)}(n+1) \leftarrow g^{(l)}(n) + \mu b(n+1)$
19: $w_k^{(l)}(n+1) \leftarrow w_k^{(l)}(n) - \eta F^{-1} g^{(l)}(n+1)$
20: **end for**
21: **end for**
22: **end for**

Принцип работы обратного распространения ошибок для сверточных нейронных сетей аналогичен Алгоритму 5. Отличие в том, что, помимо полносвязных слоев, необходимо учитывать слои свертки и пулинга при регулировании значений весов. Известно, что операция свертки [19] проводится следующим образом:

$$(I * K)_{i,j} = \sum_{p=0}^{k_1-1} \sum_{q=0}^{k_2-1} I_{i+p,j+q} \cdot K_{p,q}, \tag{11}$$

где $I$ – входное изображение и $K$ – ядро размерностью $k_1 \times k_2$. Прямое распространение в сверточных нейронных сетях проводится по следующей формуле:

$$y^{(l+1)}(n) = \phi\left(w^{(l)}(n) * y^{(l)}(n)\right). \tag{12}$$

После чего проводится обратное распространение ошибок, где функция ошибок вычисляется следующим образом:

$$\frac{\partial E(n)}{\partial w_{p,q}^{(l)}(n)} = \sum_{i=0}^{H-k_1} \sum_{j=0}^{W-k_2} \frac{\partial E(n)}{\partial x_{p,q}^{(l)}(n)} \frac{\partial x_{p,q}^{(l)}(n)}{\partial w_{p,q}^{(l)}(n)}. \tag{13}$$

Подставив (13) в (10), получим формулу обратного распространения ошибки для сверточных нейронных сетей. Аналогичную подстановку можно выполнить в рекуррентных нейронных сетях. Модель рекуррентной нейронной сети похожа на модель многослойного перспетрона по строению архитектур. Основное отличие состоит в том, что рекуррентные сети могут использовать свою внутреннюю память для обработки последовательностей произвольной длины, что позволяет более точно обрабатывать временные ряды. В последнее время наибольшее распространение получили сети с долговременной и кратковременной памятью (LSTM) и управляемым рекуррентным блоком (GRU).

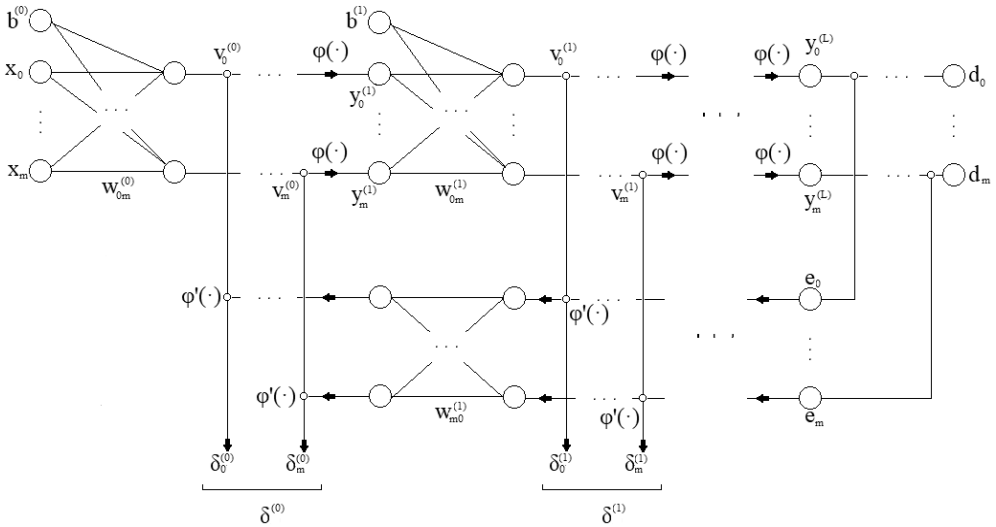


*Рис. 1. Предлагаемая схема обратного распространения ошибок с использованием натурального градиентного спуска*

Прямое распространение в рекуррентных нейронных сетях проводится следующим образом:

$$y^{(l+1)}(n) = \phi(w^{(l)}(n)\, y^{(l)}(n) + u^{(l)}(n-1)\, y^{(l)}(n-1)), \tag{14}$$

где $u^{(l)}(n-1)$ – матрица весов с предыдущего состояния нейрона. После чего проводится обратное распространение ошибок, в котором локальные градиенты функции потерь вычисляются так же, как и в мо-

дели многослойного пресептрона, для весов во входном, скрытом и выходном слоях.

### *4. Экспериментальная часть*

В экспериментальной части показаны результаты работы предложенного алгоритма натурального градиентного спуска с импульсом, имеющие большую точность по сравнению с известными аналогами в задачах распознавания изображений баз MNIST и CIFAR10. Кроме того, представлены результаты прогнозирования временных рядов с помощью рекуррентных нейронных сетей, где предложенный Алгоритм 4 достиг наименьшего значения функции ошибок.

Для проведения экспериментов на базе данных MNIST (рукописные цифры от 0 до 9) в качестве тестируемых моделей выбраны многослойный персептрон и сверточная нейронная сеть LeNet 5 из рис. 2. Нейронные сети данных архитектур не требуют много времени для обучения, и их точность зависит от метода оптимизации.

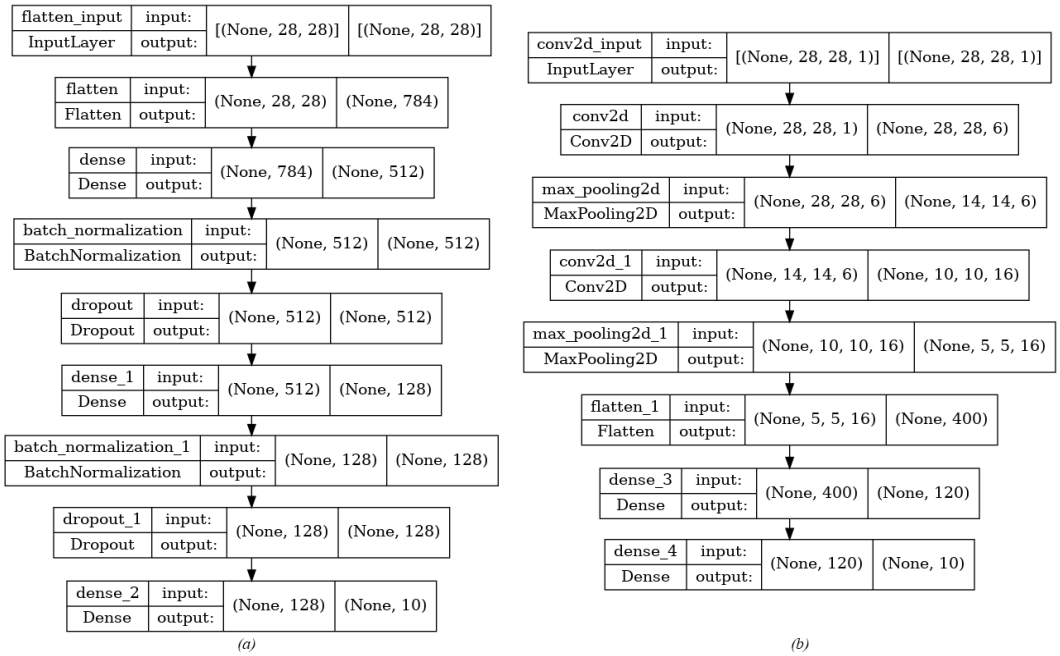


*Рис. 2. Архитектуры многослойного персептрона (a) и LeNet 5 (b) для базы изображений MNIST*

Уточним, что None на рис. 2 означает возможность выбора сета (batch) произвольного размера. Flatten – слой, выравнивающий вход. Dense – полносвязный слой. BatchNormalization – нормализация слоев для ускорения и устойчивости распознавания. Conv2d – двумерная свертка. MaxPooling2D – выборка максимального значения из карты признаков. Dropout – регуляризатор, решающий проблему переобучения.

Реализация экспериментов проводилась на Python 3.8.1 с библиотекой машинного обучения TensorFlow 2.8. Ввиду небольшой архитектуры многослойного персептрона для матрицы Фишера распределения Дирихле были выбраны параметры $\alpha_i = 1{,}4 + 0{,}005^i$, $i \in \mathbb{N}$. Для матрицы Фишера обобщенного распределения Дирихле были выбраны значения параметров $\alpha_i = 3{,}5 + 0{,}005^i$, $\beta_i = 3{,}9 + 0{,}005^i$, $i \in \mathbb{N}$. С такими значениями предложенные алгоритмы достигают наибольшей точности, причем для NGDM Dir и NGDM GenDir была выбрана скорость обучения, равная 0,1.

В случае сверточной нейронной сети LeNet 5 выбраны значения $\alpha_i = 8{,}8 - 0{,}005^i$ для распределения Дирихле и значения $\alpha_i = 6{,}8 + 0{,}005^i$, $\beta_i = 5{,}5 + 0{,}005^i$ для обобщенного распределения Дирихле, где $i \in \mathbb{N}$. В данном случае для предложенных алгоритмов оптимизации была выбрана скорость обучения со значением 0,001, так как сеть LeNet 5 содержит сверточные, пулинговые и полносвязные слои, где при большой скорости обучения функция потерь не минимизируется.

Для сравнения эффективности алгоритмов оптимизации были обучены многослойный персептрон и сверточная нейронная сеть LeNet 5, результаты которых представлены в табл. 1 и 2, где продемонстрированы конечные точности распознавания.

*Табл. 1. Точность и значение функции потерь многослойного персептрона на базе данных MNIST*

| Параметр | Алгоритмы оптимизации | | | | |
|---|---|---|---|---|---|
| | Известные | | | Предложенные | |
| | SGDM | Adam | Apollo | NGDM (Dir) | NGDM (GDir) |
| Точность (%) | $98{,}00 \pm 0{,}04$ | $98{,}01 \pm 0{,}02$ | $98{,}03 \pm 0{,}01$ | $\mathbf{98{,}15 \pm 0{,}01}$ | $98{,}12 \pm 0{,}01$ |
| Функция потерь | $0{,}1055 \pm 0{,}02$ | $0{,}0809 \pm 0{,}01$ | $0{,}0844 \pm 0{,}005$ | $\mathbf{0{,}0777 \pm 0{,}003}$ | $0{,}0799 \pm 0{,}003$ |

*Табл. 2. Точность и значение функции потерь LeNet 5 на базе данных MNIST*

| Параметр | Алгоритмы оптимизации | | | | |
|---|---|---|---|---|---|
| | Известные | | | Предложенные | |
| | SGDM | Adam | Apollo | NGDM (Dir) | NGDM (GenDir) |
| Точность (%) | 98,91 ± 0,03 | 98,99 ± 0,03 | 99,0 ± 0,001 | **99,12 ± 0,03** | 99,11 ± 0,03 |
| Функция потерь | 10,421 ± 1,08 | 7,0309 ± 0,05 | 7,6442 ± 0,08 | **5,3276 ± 0,03** | 5,3703 ± 0,02 |

Для проведения экспериментов на базе данных CIFAR10 выбрана сверточная нейронная сеть, представленная на рис. 3. Данная архитектура подходит для сравнения методов оптимизаций, но за счет операций свертки будет работать дольше. Для такой базы данных, как CIFAR10, подобная архитектура является относительно быстрой.

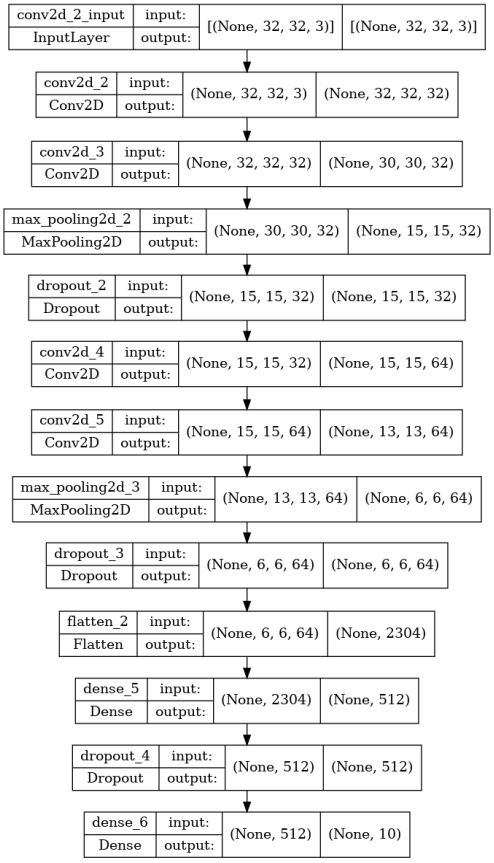


*Рис. 3. Сверточная нейронная сеть для базы данных CIFAR10*

Для сверточной сети, распознающей изображения из CIFAR10, были выбраны параметры для распределения Дирихле $\alpha_i = 7{,}4 - 0{,}02^i$, для обобщенного распределения Дирихле были выбраны $\alpha_i = 7{,}4 - 0{,}05^i$, $\beta_i = 7{,}6 - 0{,}01^i$, $i \in \mathbb{N}$. Данная сеть содержит больше слоев, чем LeNet 5, вследствие чего была выбрана скорость обучения, равная 0,004. Проведем обучение сверточной нейронной сети и, как в случае многослойного персептрона и сверточной сети LeNet 5, продемонстрируем результаты точности распознавания и минимизации функции ошибок в табл. 3.

Помимо улучшения распознаваний изображений, NGDM Nesterov с распределениями Дирихле способен успешно обрабатывать данные временных рядов. Например, сделать прогноз дальнейшего поведения зашумленного сигнала, который представляет из себя синусоиду на рис. 4.

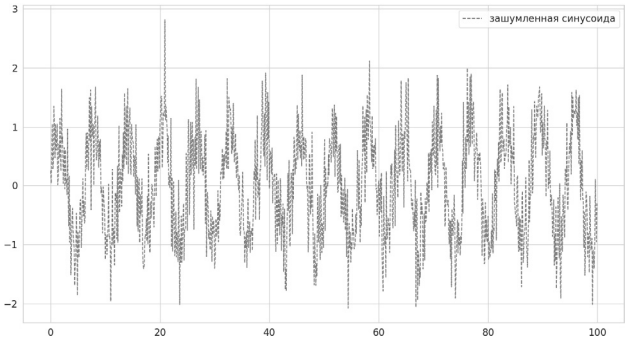


*Рис. 4. Зашумленная синусоида для эксперимента прогнозирования временных рядов*

Задача рекуррентной нейронной сети – спрогнозировать поведение сигнала на дальнейших промежутках, который визуально будет более приближен к обычной синусоиде. Для решения данной задачи использованы рекуррентные нейронные сети, представляющие из себя персептроны со слоями LSTM и GRU, состоящие из 128 нейронов с функцией активации гиперболического тангенса. На рис. 5 и 6 представлены прогнозы временного ряда, полученные нейронными сетями со слоями LSTM и GRU соответственно, и реальное поведение сигнала.

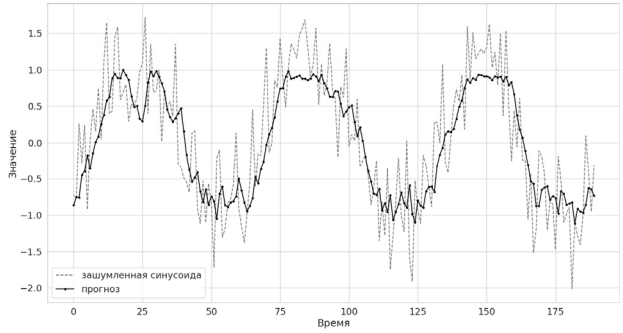


*Рис. 5. График прогноза, полученный с помощью слоев LSTM*

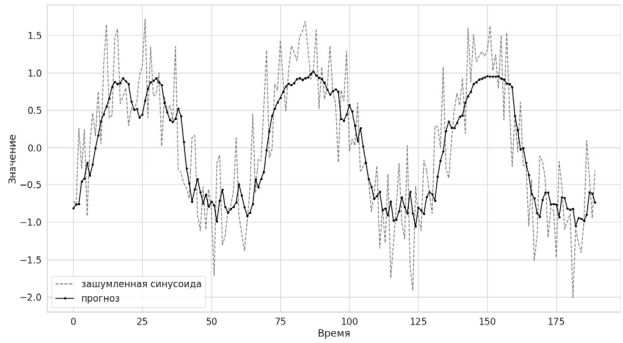


*Рис. 6. График прогноза, полученный с помощью слоев GRU*

При обучении рекуррентных сетей, прогнозирующих временной ряд зашумленной синусоиды (рис. 2), для NGDM с распределением Дирихле со скоростью обучения, равной 0,007, были выбраны параметры $\alpha_i = 4{,}4 - 0{,}005^i$, для обобщенного распределения Дирихле – $\alpha_i = 4{,}4 - 0{,}005^i$, $\beta_i = 5{,}4 - 0{,}005^i$, $i \in \mathbb{N}$. Представим результаты минимизации функции потерь (средней квадратичной ошибки) после обучения рекуррентных сетей со слоями LSTM и GRU.

Из табл. 4 можно видеть, что предложенные NGDM Dir и NGDM GenDir минимизируют функцию потерь с наибольшей точностью. Среди известных алгоритмов самым точным оказался Adam, затем SGDM Nesterov. Худший результат показал Apollo.

*Табл. 3. Точность и значение функции потерь нейронной сети из рис. 3 на базе данных CIFAR10*

| Параметр | Алгоритмы оптимизации | | | | |
|---|---|---|---|---|---|
| | Известные | | | Предложенные | |
| | SGDM | Adam | Apollo | NGDM (Dir) | NGDM (GenDir) |
| Точность (%) | $64{,}70 \pm 2{,}45$ | $77{,}90 \pm 0{,}2$ | $75{,}94 \pm 0{,}43$ | $\mathbf{78{,}49 \pm 0{,}2}$ | $78{,}43 \pm 0{,}16$ |
| Функция потерь | $1{,}0376 \pm 0{,}3$ | $0{,}6619 \pm 0{,}1$ | $0{,}7025 \pm 0{,}1$ | $\mathbf{0{,}6338 \pm 0{,}05}$ | $0{,}6363 \pm 0{,}05$ |

*Табл. 4. Значение функции потерь рекуррентных нейронных сетей со слоями LSTM и GRU при обучении на зашумлённом синусоидальном временном ряде*

| Архитектуры | Алгоритмы оптимизации | | | | |
|---|---|---|---|---|---|
| | Известные | | | Предложенные | |
| | SGDM | Adam | Apollo | NGDM (Dir) | NGDM (GenDir) |
| LSTM | $0{,}3201 \pm 10^{-6}$ | $0{,}3074 \pm 10^{-6}$ | $0{,}3245 \pm 10^{-5}$ | $\mathbf{0{,}2825 \pm 10}^{-6}$ | $0{,}2939 \pm 10^{-6}$ |
| GRU | $0{,}3243 \pm 10^{-5}$ | $0{,}3079 \pm 10^{-6}$ | $0{,}3299 \pm 10^{-5}$ | $\mathbf{0{,}2937 \pm 10}^{-6}$ | $\mathbf{0{,}2937 \pm 10}^{-6}$ |

Наличие разности в $0{,}005^i$ не дает матрице Фишера становиться единичной, так как в этом случае натуральный градиент становится стохастическим.

Основываясь на результатах, собранных в табл. 1–4, можно сделать выводы, что предложенный метод натурального градиентного спуска ускорил процесс обучения нейронных сетей и достиг наибольшей точности распознавания образов и прогнозирования временных рядов.

### *Заключение*

Предложенный метод натурального градиентного спуска с импульсом на основе распределений Дирихле оптимизирует функцию потерь быстрее и точнее, по сравнению со стохастическим градиентным спуском, адаптивной оценкой момента и адаптивным по параметрам диагональным квазиньютоновским методом невыпуклой стохастической оптимизации. Основное преимущество NGDM Nesterov с распределениями Дирихле состоит в том, что данный подход учитывает не только направление градиентов, но, как и алгоритм Apollo, учитывает выпуклость минимизируемой функции, в отличие от SGDM Nesterov и Adam. Предложенный подход, по сравнению с Apollo, не аппроксимирует Гессиан минимизируемой функции, а заменяет его на информационную матрицу Фишера, которая при использовании распределений Дирихле является постоянной. Так как матрицу Фишера не нужно пересчитывать каждую итерацию, количество вычислений уменьшается и повышается скорость обучения. Следовательно, применение предложенного метода оптимизации в различных архитектурах нейронных сетей ускорит процесс распознавания образов и прогнозирования временных рядов, достигая высокой точности.

В дальнейших исследованиях планируется внедрение метода натурального градиентного спуска с распределениями Дирихле в сверточных нейронных сетях по типу AlexNet, VGG16, SqueezeNet, GoogLeNET и ResNet-101. Сети такой архитектуры способны распознавать изображения любой собранной базы данных, а внедрение предложенного метода оптимизации функции потерь даст возможность повысить точность в процессе обучения, затрачивая меньше времени.

Стоит отметить, что на основе натурального градиента также разрабатываются подходы к реализации квантового машинного обучения с интенсивным использованием квантовых вычислений. На их основе был выведен квантовый натуральный градиент, который отличается от натурального градиента метрикой Фишера–Рао, состоящей из вероятностных векторов. Эту метрику на комплексном Гильбертовом пространстве еще называют обучающей метрикой Фубини (Fubini-Study metric). Данный подход способен ускорить процесс оптимизации функции потерь еще сильнее, чем обычный натуральный градиентный спуск. Развитие данной темы в дальнейших исследованиях позволит ускорить процесс обучения сверточных нейронных сетей, внедрить натуральный градиентный спуск на комплекснозначные нейронные сети и развивать обработку сигналов и изображений с помощью квантовых вычислений.

### *Благодарности*

Авторы выражают благодарность СКФУ за поддержку в рамках проекта поддержки малых научных групп и отдельных ученых. Исследование в параграфе 2 проведено в Северо-Кавказском центре математических исследований в рамках соглашения с Министерством науки и высшего образования Российской Федерации

(соглашение № 075-02-2022-892). Исследование в параграфе 2 проведено при поддержке Российского научного фонда (проект № 21-71-00017). Исследование в параграфе 3 проведено при поддержке Российского научного фонда (проект № 22-71-00009).

### *References*

[1] Gardner WA. Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique. Signal Proces 1984; 6(2): 113-133. DOI: 10.1016/0165-1684(84)90013-6.

[2] Loizou N, Richtárik P. Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods. Comput Optim Appl 2020; 77: 653-710. DOI: 10.1007/s10589-020-00220-z.

[3] Gao S, Pei Z, Zhang Y, Li T. Bearing fault diagnosis based on adaptive convolutional neural network with Nesterov momentum. IEEE Sens J 2021; 21(7): 9268-9276. DOI: 10.1109/JSEN.2021.3050461.

[4] Hadgu AT, Nigam A, Diaz-Aviles E. Large-scale learning with AdaGrad on Spark. 2015 IEEE Int Conf on Big Data (Big Data) 2015: 2828-2830. DOI: 10.1109/BigData.2015.7364091.

[5] Wang Y, Liu J, Mišić J, Mišić VB, Lv S, Chang X. Assessing optimizer impact on DNN model sensitivity to adversarial examples. IEEE Access 2019; 7: 152766-152776. DOI: 10.1109/ACCESS.2019.2948658.

[6] Xu D, Zhang S, Zhang H, Mandic DP. Convergence of the RMSProp deep learning method with penalty for nonconvex optimization. Neural Netw 2021; 139: 17-23. DOI: 10.1016/j.neunet.2021.02.011.

[7] Melinte DO, Vladareanu L. Facial expressions recognition for human–robot interaction using deep convolutional neural networks with rectified Adam optimizer. Sensors 2020; 20: 2393. DOI: 10.3390/s20082393.

[8] Noh S-H. Performance comparison of CNN models using gradient flow analysis. Informatics 2021; 8: 53. DOI: 10.3390/informatics8030053.

[9] Huang Y, Zhang Y, Chambers JA. A Novel Kullback–Leibler divergence minimization-based adaptive student's t-filter. IEEE Trans Signal Process 2019; 67(20): 5417-5432. DOI: 10.1109/TSP.2019.2939079.

[10] Asperti, A. Trentin. M. Balancing reconstruction error and Kullback-Leibler divergence in variational autoencoders. IEEE Access 2020; 8: 199440-199448. DOI: 10.1109/ACCESS.2020.3034828.

[11] Martens J. New insights and perspectives on the natural gradient method. J Mach Learn Res 2020; 21(146): 1-76.

[12] Ma X. Apollo: An adaptive parameter-wise diagonal quasi-newton method for nonconvex stochastic optimization. arXiv Preprint. 2021. Source: ⟨https://arxiv.org/abs/2009.13586⟩.

[13] Li W, Montúfar G. Natural gradient via optimal transport. Information Geometry 2018; 1: 181-214. DOI: 10.1007/s41884-018-0015-3.

[14] Alvarez F, Bolte J, Brahic O. Hessian Riemannian gradient flows in convex programming. SIAM 2004; 43(2): 68-73. DOI: 10.1137/S0363012902419977.

[15] Abdulkadirov RI, Lyakhov PA. Improving extreme search with natural gradient descent using Dirichlet distribution. In Book: Tchernykh A, Alikhanov A, Babenko M, Samoylenko I, eds. Mathematics and its applications in new computer systems. Cham: Springer Nature Switzerland AG; 2022: 19-28. DOI: 10.1007/978-3-030-97020-8_3.

[16] Graf M. Regression for compositions based on a generalization of the Dirichlet distribution. Stat Methods Appt 2020; 29: 913-936. DOI: 10.1007/s10260-020-00512-y.

[17] Li Y. Goodness-of-fit tests for Dirichlet distributions with applications. A PhD dissertation. 2015.

[18] Haykin SS. Neural networks: a comprehensive foundation. Prentice Hall; 1999.

[19] Aghdam HH, Heravi EJ. Guide to convolutional neural networks: A practical application to traffic-sign detection and classification. Cham: Springer International Publishing AG; 2017.

### *Сведения об авторах*

**Абдулкадиров Руслан Ибрагимович**, 2000 года рождения, студент Северо-Кавказского федерального университета с 2018 года по специальности «Прикладная математика и информатика», лаборант Северо-Кавказского центра математических исследований. Область научных интересов: машинное обучение, функциональный анализ. E-mail: *ruslanabdulkadirovstavropol@gmail.com*.

Сведения об авторе **Ляхов Павел Алексеевич** см. стр 78 этого номера.

*ГРНТИ: 28.23.15*

*Поступила в редакцию 7 апреля 2022 г. Окончательный вариант – 24 августа 2022 г.*

# A new approach to training neural networks using natural gradient descent with momentum based on Dirichlet distributions

*R.I. Abdulkadirov [1], P.A. Lyakhov [2]*
*[1] North-Caucasus Center for Mathematical Research, 355009, Russia, Stavropol, Pushkin str. 1;*
*[2] North-Caucasus Federal University, 355009, Russia, Stavropol, Pushkin str. 1*

***Abstract***

In this paper, we propose a natural gradient descent algorithm with momentum based on Dirichlet distributions to speed up the training of neural networks. This approach takes into account not only the direction of the gradients, but also the convexity of the minimized function, which significantly accelerates the process of searching for the extremes. Calculations of natural gradients based on Dirichlet distributions are presented, with the proposed approach introduced into an error backpropagation scheme. The results of image recognition and time series forecasting during the experiments show that the proposed approach gives higher accuracy and does not require a large number of iterations to minimize loss functions compared to the methods of stochastic gradient descent, adaptive moment estimation and adaptive parameter-wise diagonal quasi-Newton method for nonconvex stochastic optimization.

*Keywords*: pattern recognition, machine learning, optimization, Dirichlet distributions, natural gradient descent.

*Citation*: Abdulkadirov RI, Lyakhov PA. A new approach to training neural networks using natural gradient descent with momentum based on Dirichlet distributions. Computer Optics 2023; 47(1): 160-169. DOI: 10.18287/2412-6179-CO-1147.

*Acknowledgements*: The authors would like to thank the North-Caucasus Federal University for the award of funding in the contest of competitive projects of scientific groups and individual scientists of the North-Caucasus Federal University. The research in section 2 was supported by the North-Caucasus Center for Mathematical Research through the Ministry of Science and Higher Education of the Russian Federation (Project No. 075-02-2022-892). 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).

***Authors' information***

**Ruslan Ibragimovich Abdulkadirov** (b. 2000) is a student of the North-Caucasus Federal University since 2018 with a degree in Applied Mathematics and Informatics, works as a laboratory assistant at the North-Caucasus Center for Mathematical Research. Research interests: machine learning, functional analysis.
E-mail: *ruslanabdulkadirovstavropol@gmail.com* .

**Pavel Alekseevich Lyakhov** (b. 1988) graduated from Stavropol State University, specialty "Mathematics" in 2009. PhD of Physical and Mathematical Sciences. Head of the Department of Mathematical Modeling, North-Caucasus Federal University. Research interests are digital signal and image processing, artificial intelligence, neural networks, modular arithmetic, parallel computing, high-performance computing, digital circuits and hardware accelerators.
E-mail: *ljahov@mail.ru* .

*Received April 7, 2022. The final version – August 24, 2022.*