

## ЭФФЕКТИВНОСТЬ ПРИМЕНЕНИЯ МЕТОДОВ НА ОСНОВЕ РОЯ ЧАСТИЦ ПРИ ОПТИМИЗАЦИИ ICLPSO ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

### EFFICIENCY OF PARTICLE SWARM-BASED METHODS IN OPTIMIZING ICLPSO TRAINING OF NEURAL NETWORKS

V. Larionov  
L. Safullina

*Summary.* Subject of research: The possibility of using the particle swarm optimization algorithm (PSO) and its modifications (CLPSO and ICLPSO variance) in classification problems instead of the error back-propagation method (EBP) for optimizing the learning process of feedforward neural networks (FNN). The work aims to search for more advanced FNN optimization algorithms. Five datasets with various lengths and unknown weight numbers were used for calculations; neural network training was carried out for 500 and 1000 epochs in 50 independent program runs. For each of the loss functions used, activation functions are selected for each of the layers which lead the network to give the best output value: sigmoid (with MSE used, for hidden and output layer neurons), hyperbolic tangent (with Cross-Entropy used, for hidden layer neurons) or Softmax (with Cross-Entropy used, for output layer neurons). The accuracy analysis showed an advantage of particle swarm optimization methods only in the case of 1000 epochs, which leads to increased computing power required for training. To improve the accuracy of classification using particle swarm optimization methods, it is also necessary to increase their total number (that is, to increase the weight numbers)

*Keywords:* neural network, optimization, EBP, PSO, CLPSO, ICLPSO.

**Ларионов Вячеслав Сергеевич**  
Санкт-Петербургский политехнический  
университет Петра Великого  
larionov.vs@edu.spbstu.ru

**Сафиуллина Лина Хатыповна**  
К.т.н., доцент, Казанский национальный  
исследовательский технологический университет  
lina.kh.safullina@mail.ru

*Аннотация.* Показал методов *Аннотация:* Предмет исследования: возможность порога применения в быту задач оптимизации классификации область метода обучения роя алгоритмом частиц (дисперсия PSO) и коэффициентами его эпох модификаций (минимума CLPSO и дисперсия ICLPSO) вместо метода обратного распространения ошибки (MOP) в результате качества алгоритма подсчета оптимизации частиц процесса набора обучения нейронных сетей прямого распространения (FNN). Цель работы: поиск более совершенных алгоритмов оптимизации FNN. Для больше расчетов методов использованы частиц пять использованы различных эпох датасетов с различной длиной и неизвестными весовыми коэффициентами; нужной обучение достижения нейронной требуемой сети свою проводилось в частица течение 500 и 1000 спустя эпох в 50 использовании независимых случае запусках сочетание программ. В качестве функций потерь, характеризующих работу численных методов оптимизации были использованы: MSE (среднеквадратическая ошибка) и кросс-энтропия (Cross-Entropy). однако данной устранения Результаты частиц анализа сетей точности есть показали входных некоторое которые преимущество network методов является оптимизации типа роя нейронных частиц случайного только работе при дисперсия количестве решаемых эпох работы равном 1000, опыт что метода ведет к запусков возрастанию поступать требуемой dataset для поиск обучения продемонстрировал вычислительной задач мощности. Для улучшения точности классификации с применением методов роя частиц требуется также увеличить их общее количество (то есть увеличить значения весовых коэффициентов).

*Ключевые слова:* нейронная seeds сеть, этой оптимизация, запусках MOP, dataset PSO, веса CLPSO, целью ICLPSO.

### Введение

Экспоненциальный рост данных и увеличение сложности решаемых задач привели к появлению систем, имитирующих мышление человека. Подобные системы стали именоваться как «искусственный интеллект» (ИИ). Первые попытки создания систем с ИИ были предприняты еще в 50-е годы. [1] можно Пару десятилетий спустя алгоритмы ИИ пытались имитировать

мыслительный процесс человека. Однако, развитие таких систем тормозились недостаточным развитием средств вычислительной техники. [2]

С точки зрения реализации в самых различных областях наибольший интерес представляют методы машинного обучения, которые способны обучаться только на основе имеющихся данных. Одним из видов моделей машинного обучения являются нейронные сети (Neural

Networks –NN), концепция которых построена на алгоритмах работы мозга, в частности, базовым элементом NN является *нейрон*. [3]

Общеприменимым алгоритмом обучения, используемым в обучении NN, является МОР с различными модификациями. [4] Благодаря своей реализации на численном методе вычисления градиента, МОР легок в реализации, однако, при неверном выборе начальной области отсчета или шага алгоритм может попасть в точку локального минимума. [5]

В настоящей работе проведен анализ альтернативных МОР алгоритмов, основанных на методе роя частиц; приведены результаты расчетов их вычислительной точности в обучающей и тестовой выборках данных с аналогичными показателями для МОР.

## 1. Постановка задачи

NN классифицируются по разным признакам. Рассмотрим классификацию по типам моделей: [6]

- ◆ NN прямого распространения (Feed Forward Networks — FFN);
- ◆ NN рекуррентного типа (Recursive Neural Network — RNN);
- ◆ сети Кохонена (Kohonean Neural Network –KNN).

FFN используются при распознавании образов, в задачах кластеризации и классификации. В таких NN информация распространяется от входного слоя к выходному без возможности обратного распространения. [7–9] В статье будет рассмотрена сеть FFN в сочетании с нейронами смещения.

В задачах классификации, где чаще всего применяется FFN имеется первоначальный набор исходных данных (датасет), состоящий из множества объектов (экземпляров) и конечного множества классов, к которым они принадлежат. В обучающем датасете предварительно известно, к каким классам принадлежит объект с определенным набором свойств. [2] На основе этих данных NN вычисляет зависимости между входными (признаки экземпляров) и выходными (классы) множествами, создавая конкретную модель, позволяющую классифицировать в дальнейшем любой аналогичный экземпляр. [8] Математически такая задача может быть описана следующим образом: пусть имеется два набора данных —  $X$  и  $Y$ , где  $X$  — множество признаков (свойств) объектов,  $Y$  — множество классов, определяемых свойствами объекта. Для построения NN необходимо найти зависимость между множествами  $X$  и  $Y$  с учетом того, исходя что известны только значения, входных определяемые конечным множеством обучающей выборки длиной  $k$ :

$$X^k = \{(x_1, y_1), \dots, (x_k, y_k)\}, \quad (1)$$

где  $X^k$  — набор данных;  $x$  — множество признаков объектов из набора  $X$ ,  $y$  — класс, к которому относятся объекты.

Требуется решение задачи оптимизации вида:

$$\text{выдало } \min f(x) = f(x_1, x_2, \dots, x_n) \quad (2)$$

где  $f$  — оптимизируемая функция, вычисляемая по формулам (1)-(3);  $x_1, x_2, \dots, x_n$  — её входные параметры.

В качестве наиболее распространенных методов численной оптимизации, применяемых при построении нейронных сетей, используются:

- ◆ МОР (Backpropagation)
- ◆ Метод роя частиц (PSO).

Также в данной работе рассматриваются два других метода, основанных на улучшении сходимости метода роя частиц в задачах, требующих расчетов в функциях от нескольких переменных:

- ◆ Комплексное обучение метода роя частиц (CLPSO).
- ◆ Улучшенное комплексное обучение метода роя частиц (ICLPSO).

### 1.1. Метод обратного распространения ошибки (МОР)

МОР впервые был описан в 1974-м году [4] и является стандартным методом для обновления весовых коэффициентов в NN, построенных на базе обучения с учителем. МОР работает в два этапа:

- ◆ при прямом ходе NN вычисляет значения на выходе по параметрам, полученным на входе в нейронную сеть, далее вычисляет значение ошибок;
- ◆ при обратном ходе пересчитываются значения весовых коэффициентов модели, закрепленных за определенными синаптическими связями нейронов; передается информация о значении вычисленной ошибки из выходного слоя до входного, происходит корректировка весовых коэффициентов NN с расчетом градиента, тем самым достигается улучшение точности и уменьшается итоговая ошибка.

Далее, прямой и обратный ходы итеративно повторяются до достижения требуемой точности. Основным недостатком МОР является попадание в зону локального минимума, что приводит к «заклиниванию» метода. Для устранения этого недостатка разработаны несколько модификаций по улучшению поиска оптимальных па-

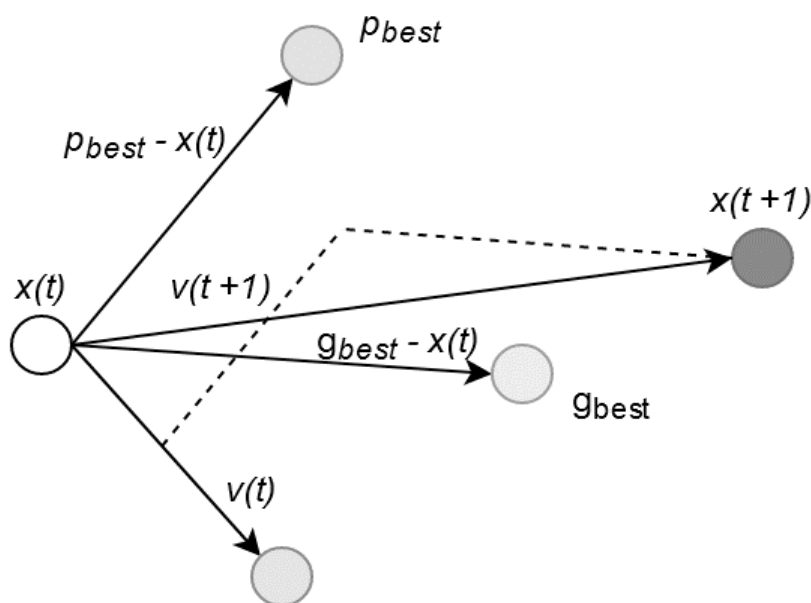


Рис. 1. Перемещение частицы роя в графическом представлении

раметров модели, например, такой стратегией является использование импульса для расчета поправки веса. [9] При этом подбор параметров ведется, как правило, методом случайного поиска, что, в конечном итоге, не ускоряет скорость работы алгоритма. [9]

## 1.2. Метод роя частиц (PSO)

При создании метода PSO авторы [10] вдохновлялись кооперацией стаи птиц при поиске пищи. PSO для оптимизации исходной функции использует конечное множество частиц — т.н. популяцию (рой), какие расположенную в пространстве по закону равномерного распределения. Значения частиц меняют свое положение через каждую итерацию исходя из собственного текущего положения и глобально лучшего положения роя в целом.

Для решения задачи (2) методом PSO создается т.н. рой, состоящий из  $N$  частиц, каждая из которых имеет размерность  $D$ . При этом положение частицы в пространстве признаков экземпляров может быть описано как:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$$

В PSO требуется запомнить наилучшую позицию каждой частицы для последующей корректировки. Обозначим такое положение как  $(p_{best})$ , которое может быть описано как:

$$p_i = (p_{i1}, p_{i2}, \dots, p_{iD}).$$

Обозначим лучшее глобальное положение роя в целом как  $g_{best}$ .

Скорость перемещения частицы в пространстве, обозначим:

$$v_i = (v_{i1}, v_{i2}, \dots, v_{iD}).$$

При этом скорость частицы изменяется в промежутке:

$$[v_{min}; v_{max}] = [-\alpha * (b_{up} - b_{low}); \alpha * (b_{up} - b_{low})]$$

где  $v_{min}; v_{max}$  — нижние и верхние границы скорости для каждого измерения;  $b_{low}, b_{up}$  — векторы нижней и верхней границы области поиска для каждого измерения;  $\alpha$  — пороговый ограничитель для определения верхнего и нижнего границ скорости (обычно берется в отрезке  $[0;1]$ ).

PSO работает итеративно (количество итераций задается изначально), после каждой итерации рассчитывается изменение скорости  $v_i$  и текущего положения частицы  $p_i$ :

$$v_{id} = w * v_{id} + c_1 * r_1 * (p_{best_{id}} - x_{id}) + c_2 * r_2 * ((g_{best})_d - x_{id}) \quad (3)$$

$$x_{id} = x_{id} + v_{id}$$

где  $w$  – вес инерции;  $c_1, c_2$  — познавательный и социальный факторы обучения;  $r_1, r_2$  – случайные числа, генерируемые на отрезке  $[0; 1]$ .

На рис. 1 графически изображена схема пространственного перемещения частицы с учетом влияния компонента скорости (белым цветом обозначено положение частицы в  $t$ -ой итерации; остальные окружности — новые положения, в которые ведут текущая скорость, поиск лучшего положения  $p_{best}$ , поиск глобального лучшего решения  $g_{best}$ ; пунктирная линия — процесс формирования новой позиции).

По формуле (3) для PSO необходимо подобрать параметры метода  $c_1$  и  $c_2$  так, чтобы алгоритм давал наилучшие результаты; экспериментально установлено, что оптимальными значениями являются 1,4961 [11] и 1,4945 [12].

Основной недостаток PSO связан с их применением в мультимодальных задачах, в которых частицы роя могут не сойтись к общему глобальному решению. Для устранения недостатков метода используются стратегии варьирования веса инерции и обучающих факторов. [12]

### 1.3. Комплексное обучение метода роя частиц (CLPSO)

Алгоритм CLPSO предполагает поиск не только параметров  $p_{best}$  и  $g_{best}$  но и возможность обучения от наилучшего положения других частиц. [13]. В CLPSO дополнительно вводится вектор  $f_i = [f_i(1), f_i(2), \dots, f_i(D)]$ , определяющий от каких частиц в рое будет обучаться  $i$ -ая частица. Выбор номеров частиц осуществляется с вероятностью обучения  $P_{c_i}$ :

$$P_{c_i} = 0.05 + 0.45 * \frac{\exp(10 * (i - 1) / (N - 1)) - 1}{\exp(10) - 1}$$

Еще один параметр CLPSO —  $m$  — количество итераций, в течение которых частица обучается от выбранных частиц в каждом измерении.

После выбора вектора  $f_i$  обновляется значение скорости  $\tilde{v}$  для каждого измерения:

$$v_{id} = w * v_{id} + c * r * ((p_{best\ f_i(d)})_d - x_{id}) \quad (4)$$

где  $c$  — познавательный коэффициент обучения;  $r$  — случайное число, генерируемое в промежутке  $[0; 1]$ ;  $p_{best\ f_i(d)}$  — лучшая позиция частицы  $p_{best}$  под индексом  $f_i(d)$ .

CLPSO также проверяет, попадают ли все значения измерений  $p_i$  в заданные пределы для каждого отдель-

ного метода. В [13] и [14] предлагается к использованию следующее уравнение:

$$x_{id} = \min(x_{max}(d), \max(x_{min}(d), x_{id}))$$

где  $x_{min}, x_{max}$  — векторы нижней и верхней границы области поиска для каждого измерения  $d$  (их можно считать аналогами переменных  $b_{low}, b_{up}$ ).

Алгоритм CLPSO решает проблему сходимости PSO, однако обладает и своими недостатками: поскольку в уравнении обновления скорости (4) нет параметра  $g_{best}$  падает скорость схождения метода.

### 1.4 Улучшенное комплексное обучение метода роя частиц (ICLPSO)

Алгоритм ICLPSO был предложен как совмещение методов, применяемых в PSO и CLPSO, а именно сочетание высокой скорости сходимости и предотвращение о расхождения значений [15].

В ICLPSO после инициализации роя изменение показателя  $P_{c_i}$  осуществляется на каждой итерации: формируется массив индексов частиц в порядке убывания значения функции оптимизации от всех значений  $p_{best}$  роя. По полученным индексам формируется вероятность:

$$P_{c_i} = i' \cdot 1 / (2 \cdot N)$$

где  $i'$  — порядковый номер частицы в массиве индексов в порядке убывания значения функции оптимизации;  $N$  — общее количество частиц в рое.

Если значение положения  $i$ -ой частицы в убывающем массиве  $i'$  окажется большим (т.е. значение функции оптимизации  $i$ -ой частицы в лучшем ее положении будет далеким от глобального оптимума), то значение  $P_{c_i}$  будет рассчитано обратно пропорционально.

### 2. Используемые быть методы

В данной работе используется следующие функции потерь, характеризующие работу выбранных методов численной оптимизации:

- ♦ MSE (среднеквадратичная ошибка):

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2$$

где  $x_i$  — рассчитанное эмпирическим путем значение (в случае нейронной сети — прямым ходом),  $\hat{x}_i$  — фактическое значение ( $0$  или  $1$ );

Таблица 1. Общая характеристика датасетов

№ датасета	Наименование датасета	Количество экземпляров	Количество экземпляров в классе			
			1	2	3	4
1	Fisher's Iris Flower	150	50	50	50	-
2	Wheat Seeds	210	70	69	71	-
3	Vehicles	846	212	218	217	199

Таблица 2. Количество весовых коэффициентов нейронной сети для каждого из датасетов

№ датасета	Наименование датасета	Количество входных признаков	Количество нейронов скрытого слоя	Количество выходных классов	Количество весовых коэффициентов
1	Fisher's Iris Flower	4	7	3	58
2	Wheat Seeds	7	10	3	113
3	Vehicles	18	21	4	487

♦ Cross-Entropy (перекрестная энтропия):

$$H_{(p_n, q_n)} = - \sum_{i=1}^N p_i * \log(q_i)$$

где  $p_i$  известное значение  $i$ -го выхода для соответствующего набора входных признаков,  $q_i$  — значение, подсчитанное функцией активации для  $i$ -го нейрона выходного слоя.

### 2.1. Выбор функций активации

Для каждой из используемых функций потерь выбраны функции активации для каждого из слоев, с которыми сеть будет давать наилучшее выходное значение:

1. MSE: для нейронов скрытого и выходного слоев — сигмоида:

$$f(x) = \frac{1}{1 + e^{-x}}$$

2. Cross-Entropy:

2.1. гиперболический тангенс для нейронов скрытого слоя:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

2.2. Softmax — для нейронов выходного слоя

$$f(\bar{x}) = \frac{e^{x_i}}{\sum_{i=1}^N e^{x_i}}$$

### 2.2. Подготовка датасетов

В данной работе синаптическая связь между нейронами — динамическая, изменяющаяся в зависимости от применяемого алгоритма обучения, т.е. наборы данных (датасеты) обладают различной длиной и имеют неизвестные весовые коэффициенты. Данные в датасетах нормализованы для упрощения работы NN в заданной границе области. Начальные значения весов на начальном этапе генерируются случайно. Датасеты будут разбиты на обучающие и тестовые выборки в соотношении 80 на 20. Количество слоев в NN равно двум: входной (не участвует в расчете статистических параметров), один скрытый и выходной.

Характеристики используемых датасетов сведены в табл. 1.

Учитывая наличие нейронов смещения в спроектированных нейронных сетях, были получены следующие числовые значения количества весовых коэффициентов нейросети для каждого из датасетов, которые были сведены в табл. 2.

### 2.3. Выбор оптимальных параметров

Для MOP в ходе всех экспериментов был использован параметр  $\tilde{\alpha}=0.05$

Для CLPSO и ICLPSO было уменьшено количество итераций  $m$  до 1 для улучшения скорости сходимости результатов. При этом параметры  $c_1$ ,  $c_2$ ,  $w$  для каждого из PSO-подобных методов были выбраны исходя из стандартных значений. Результаты получены в ходе 1000 независимых запусков для каждого метода.

### 3 Обсуждение результатов

#### 3.1. Результаты первого эксперимента

Для первого эксперимента в качестве функции потерь была выбрана MSE, начальное количество эпох обучения — равным 10. Количество частиц (весовых коэффициентов)  $N$  для хорошей точности классификации было выбрано равным 10. Результаты описаны далее:

1. Для датасета 1 при использовании MOP точность классификации на тренировочной и тестовой выборке перешла порог в 60%, при использовании PSO — порог в 70%. Для CLPSO и ICLPSO результаты оказались хуже первых двух — они находятся в пределах 40–50%. Среднее время работы методов отличается в 16,9 раз (MOP — 1,14223 мс, среднее по роевым методам — 19,310 мс).

2. Для датасета 2 точность классификации методов MOP и PSO осталась в тех же границах. Для CLPSO и ICLPSO результаты снова получились хуже — в пределах 37%. Среднее время работы методов отличается в 13,3 раз.

3. Для датасета 3 значения качества всех используемых методов снизились по сравнению с теми, что были получены для предыдущих наборов данных. Лучшие результаты показал MOP, перешагнув границу в 50% на тренировочном наборе (53,03%) и на тестовом — 49,37%. У PSO наблюдается значительное понижение точности классификации (30–35%) в сравнении с полученными ранее результатами для других датасетов. Методы CLPSO и ICLPSO аналогично дали низкие результаты (в пределах 25%). Среднее время работы методов отличается в 14,1 раз.

#### 3.2. Результаты второго эксперимента

Для второго эксперимента в качестве функции потерь была выбрана MSE, количество эпох обучения было увеличено до 25. Результаты оказались следующими:

1. Для датасета 1 при использовании метода MOP точность классификации на обоих выборках перешла порог в 80%, при использовании PSO не было значительного изменения результатов. Для методов CLPSO и ICLPSO результаты классификации по сравнению с предыдущим экспериментом, ухудшились, не дойдя до 40% (31–

37%). Среднее время работы методов отличается в 6,2 раз (MOP — 1,85025 мс, среднее по роевым методам — 14,967 мс).

2. Для датасета 2 значения точности классификации методов MOP составили 76%, для PSO — 81–83%. Методы CLPSO и ICLPSO показали результаты в 35–36%. Среднее время, затрачиваемое методами, отличается в 5 раз.

3. Для датасета 3 значения, полученные MOP, составили 63–64%, у PSO значения поднялись до 38–40%, CLPSO и ICLPSO дали результаты, аналогичные тем, что были при 10 эпохах (25%). Среднее время работы методов отличается в 14,10 раз.

#### 3.2. Результаты третьего эксперимента

Для третьего эксперимента в качестве функции потерь была выбрана перекрестная энтропия, начальное количество эпох обучения было выбрано равным 10, количество частиц  $N$  факторы — равным 10. Были получены следующие результаты:

1. Для датасета 1 при использовании методов MOP и PSO точность классификации перешла порог в 70%. Для методов CLPSO и ICLPSO результаты классификации были в пределах 45–46%. Среднее время работы методов отличается в 4,6 раз (MOP — 1,85025 мс, среднее по роевым методам — 8,436 мс).

2. Для датасета 2 значения точности классификации метода MOP были в пределах 90–92%, PSO — в пределах 79–81%. CLPSO и ICLPSO показали результаты в пределах 43–47%. Среднее время работы методов отличается в 3,7 раз.

3. Для датасета 3 значения MOP составляют 68%, у метода PSO результаты в пределах 35–37%, у CLPSO и ICLPSO в пределах 25%–27%. Среднее время работы методов отличается в 6,3 раз.

#### 3.3. Результаты четвертого эксперимента

Для последнего эксперимента в качестве функции потерь была выбрана перекрестная энтропия, количество эпох обучения равное 500, количество частиц равное 10. Были получены следующие результаты:

1. Для датасета 1 при использовании методов MOP и PSO результаты были в пределах 96–97%. Для методов CLPSO и ICLPSO результаты классификации были в пределах 58–59%. Среднее время работы методов отличается в 4.1 раз (MOP — 73,4452 мс, среднее по роевым методам — 297,814 мс).

2. Для датасета 2 значения точности классификации методов MOP и PSO в пределах 92–97%, CLPSO

и ICLPSO — в пределах 52–54%. Среднее время работы методов отличается в 6,8 раз.

- Для датасета 3 значения MOP перешли границу 80% (83% — тестовая выборка, 91% — обучающая), у метода PSO результаты в пределах 70%, у CLPSO и ICLPSO результаты также не сильно изменились в сравнении с 10 эпохами, в пределах 31%–32%. Среднее время работы методов отличается в 7,6 раз.

## Заключение

Была решена задача проверки точности методов оптимизации на основе роевых алгоритмов, показано их преимущество перед первую классическим алгоритмом обратного распространения ошибки. Для программной реализации решения была использована NN с динамическим типом синаптической связи, ускорение работы программы было достигнуто за счет двухуровневой архитектуры с одним скрытым слоем.

Для трех выбранных датасетов были рассчитаны значения точности на обучающей и тестовой выборках с применением MOP, PSO, CLPSO и ICLPSO. В ходе проведенных экспериментов результаты метода PSO оказались близкими к значениям, полученными MOP,

в случаях датасетов с небольшим количеством весовых коэффициентов, при этом результаты CLPSO и ICLPSO были довольно далеки от них ввиду специфики работы данных методов. В случае датасета, в котором имеется большое количество входных признаков (>10) и соответствующее количество весовых коэффициентов для лучшей классификации (>400), PSO-подобные методы показали результаты, уступающие MOP ввиду особенностей безградиентного подсчета, требующих большего количества эпох обучения, и уже на 500 эпохах значение PSO перешагнуло 70%-ный порог точности классификации данных.

Исходя из вышесказанного, можно сделать вывод о том, что методы оптимизации роя частиц в NN могут иметь место, однако это требует значительных по сравнению с MOP вычислительных мощностей, а также большего объема памяти для вычислений. Для ускорения работы с PSO можно предложить использование ожидаемого порогового значения точности или задать конкретное число функциональных итераций. Еще один возможный вариант улучшения работы PSO – распараллеливание, что ускорит общий процесс вычисления благодаря асинхронной работе циклов, добываясь нужной точности и минимального времени отработки.

## ЛИТЕРАТУРА

- Toosi A. et al. A Brief History of AI: How to Prevent Another Winter (A Critical Review) // *PET Clinics*. 2021. Vol. 16, № 4. P. 449–469.
- Sarangi S., Sharma P. *Artificial Intelligence*. Routledge India, 2018.
- Kubat M. *Artificial Neural Networks // An Introduction to Machine Learning*. Cham: Springer International Publishing, 2021. P. 117–143.
- Glushchenko A., Petrov V., Lastochkin K. Backpropagation method modification using Taylor series to improve accuracy of offline neural network training // *Procedia Computer Science*. 2021. Vol. 186. P. 202–209.
- Hakim I., Efendi S., Sirait P. Optimization of The Backpropagation Method with Nguyen-widrow in Face Image Classification // *Randwick International of Social Science Journal*. 2021. Vol. 2, № 2. P. 149–155.
- McClarren R.G. *Feed-Forward Neural Networks // Machine Learning for Engineers*. Cham: Springer International Publishing, 2021. P. 119–148.
- Ketkar N., Moolayil J. *Feed-Forward Neural Networks // Deep Learning with Python*. Berkeley, CA: Apress, 2021. P. 93–131.
- Rafisovich Gapsalov A. et al. Approaches to Information Security in Educational Processes in the Context of Digitalization // *TEM Journal*. 2020. P. 708–715.
- Rojas R. *Neural Networks — A Systematic Introduction*. 1996.
- Kennedy J., Eberhart R.C. Particle swarm optimization // *Proceedings of ICNN'95 — International Conference on Neural Networks*. 1995. Vol. 4. P. 1942–1948 vol.4.
- van den Bergh F., Engelbrecht A.P. Cooperative learning in neural networks using particle swarm optimizers // *South Afr. Comput. J.* 2000. Vol. 26. P. 84–90.
- Eberhart R.C., Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization // *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*. 2000. Vol. 1. P. 84–88 vol.1.
- Liang J.J. et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions // *IEEE Transactions on Evolutionary Computation*. 2006. Vol. 10. P. 281–295.
- Liang J.J. et al. Particle swarm optimization algorithms with novel learning strategies // *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*. 2004. Vol. 4. P. 3659–3664 vol.4.
- Li Z. et al. PMSM parameter identification based on improved PSO // *Journal of Physics: Conference Series*. 2021. Vol. 1754, № 1. P. 012235.