



A heuristic approach to the hyperparameters in training spiking neural networks using spike-timing-dependent plasticity

Dawid Połap¹ · Marcin Woźniak¹ · Waldemar Hołubowski¹ · Robertas Damaševičius¹

Received: 17 March 2021 / Accepted: 4 December 2021 / Published online: 16 December 2021
© The Author(s) 2021

Abstract

The third type of neural network called spiking is developed due to a more accurate representation of neuronal activity in living organisms. Spiking neural networks have many different parameters that can be difficult to adjust manually to the current classification problem. The analysis and selection of coefficients' values in the network can be analyzed as an optimization problem. A practical method for automatic selection of them can decrease the time needed to develop such a model. In this paper, we propose the use of a heuristic approach to analyze and select coefficients with the idea of collaborative working. The proposed idea is based on parallel analyzing of different coefficients and choosing the best of them or average ones. This type of optimization problem allows the selection of all variables, which can significantly affect the convergence of the accuracy. Our proposal was tested using network simulators and popular databases to indicate the possibilities of the described approach. Five different heuristic algorithms were tested and the best results were reached by Cuckoo Search Algorithm, Grasshopper Optimization Algorithm, and Polar Bears Algorithm.

Keywords Spiking neural network · Heuristic · Hyperparameters · Federated learning · Image processing

1 Introduction

Spiking neural networks are representatives of the third type of neural network, where, unlike its predecessor, it models the operation of information flow in the human body in much more detail. The last years brought much research not only about concepts and differences from different mathematical models but also the practical

application of these structures [32]. Spiking network types differ from their predecessors mainly in two elements. The first one is to add a concept of time relative to the flow of information between neurons. The second element is to remove the automatic forwarding of information from the neuron. This is solved by analyzing the voltage level in the neuron, where sending information is forced by exceeding a certain threshold value for the neuron. Remodeling and increasing the realism of neural models meant changing the whole operation. And the consequence of this is that the known training algorithms are no longer useful. There are some unsupervised approaches like spike-timing-dependent plasticity (STDP), but it is difficult to propose a supervised one. Lee et al. [17] described a proposition to develop spike-based backpropagation idea when [26] shown a training solution using equilibrium propagation. Again, Saunders et al. [30] described another unsupervised technique for feature training. Spiking networks can be shallow and as deep as their predecessors. Diehl and Cook [9] used a two-layer architecture and obtained accuracy over 90% on image datasets. In contrast, Sengupta et al. [31] analyzed a deep architectures. In both cases, it was tested that shallow and deep structures can obtain a high accuracy level in different classification tasks. The main

This work is supported by the Silesian University of Technology, Poland under Grant No. BKM-512/RMS2/2021, and by the Rector proquality grant at the Silesian University of Technology, Poland No. 09/020/RGJ21/0007 and 09/020/RGJ21/0008.

✉ Dawid Połap
Dawid.Polap@polsl.pl

Marcin Woźniak
Marcin.Wozniak@polsl.pl

Waldemar Hołubowski
Waldemar.Holubowski@polsl.pl

Robertas Damaševičius
Robertas.Damasevicius@polsl.pl

¹ Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23 44-100, Gliwice, Poland

issue is what kind of data is processed and how large is the database.

The last research brought many different architectures and modifications of these networks like a quantum version described by Kristensen et al. [14]. Developing a new structure is associated with many problems such as routing, sorting, etc. [24] presented the obtained results of research on analytical performance assessment of routing algorithm. The use of real equipment and electrodes can implement very large and deep networks, where the number of spikes can be huge, so sorting them is a very important problem [4]. Balaji et al. [2] presented an alternative way of mapping neural networks to neuromorphic hardware. In their paper, the result was compared with other, existing solutions proposed approach reduce average energy consumption and delay time. The perception of depth in vision is also a complicated problem that was presented by Haessig et al. [11]. A similar idea can be the analysis of knowledge representation that can be very useful for brain-computer interfaces and it was proved by Kumarasinghe et al. [15].

Moreover, Capecchi et al. [5] described evolving spiking neural networks for modeling gene interaction networks based on time-series gene expression. Even though the development of these techniques is in the early stages, numerous applications are already found and tested on simulators like hand gesture analysis [6] or discrimination of EMG signals [10]. An interesting idea was proposed by Rongala et al. [28], where the classification problem naturalistic texture stimuli were analyzed using spiking architecture. A similar, application can be seen in the proposal of Wang et al. [35], which analyze one of the most difficult images/video classification problem—action recognition. The number of problems that can be solved by the application of different neural networks and mathematical models is very large [8, 13, 19] and the possibility of application may improve the quality of human lives [20]. Moreover, the various aspects of neural models are used in other approaches like medicine [23], or remote sensing [36].

The very large development of spiking neural networks indicates the great practical potential and much more accurate modeling of information flow by nerve cells in living organisms. It is worth noting that such systems and architectures have a huge number of variables, the values of which are quite often chosen empirically based on conducted experiments. The automatic method can help in decreasing the time to develop the best neural structure. Based on these observations, we noticed that the selection of parameters can be reduced to an optimization problem. Particularly, it is important because of the huge development of heuristic techniques inspired by nature. One of the most popular heuristics is the cuckoo search algorithm described by Yang and Deb [39], where the main idea is

based on the behavior of cuckoos during tossing their eggs to other nests. In the last two years, there was modeled the behavior of whales [21], polar bears [27], grasshopper [29], or salp swarms [33]. Moreover, optimization methods founds in many areas of practical implementation, for instance in plant waste analysis [34] or job scheduling [7]. Theoretical aspects on hyperparameters optimization tools are shown in [38], where the mentioned meta-heuristic was found to be a good solution for finding parameters in classic machine learning tools like an artificial neural network, etc. Again, in [18] a new method for the hyperparameter problem was introduced. The author proposed a bandit-based method and analyzed it with Bayesian methods.

In this paper, we propose using heuristic algorithms for searching for the best coefficients for spiking neural networks. The proposed approach is based on the modification of the heuristic operation idea and by proposing a fitness function that can be used for hyperparameters problems with the use of the federated learning method. To summarize, the main contribution is as follows:

- New mechanism for finding hyperparameters in spiking neural networks,
- Combining collaborative learning with heuristic operations,
- The use of multithreading in hybrid (neuro-heuristic) analysis of the coefficients used in the training process,
- Evaluation of the proposal by five different heuristic algorithms inspired by nature for selecting the best one.

2 Spiking neural network

In this section, we described a mathematical model of spiking neural network and spike-timing-dependent plasticity algorithm which is used for training these structures. The analyzed model was proposed by Diehl and Cook [9].

2.1 Spiking neuron

One of the common known neuron models is called the leaky integrate-and-fire (LIF) model which takes into account the integrative properties of the membrane in the neuron as well as the leakage of current due to passive ion channels. Let us denote the membrane voltage at V . The neuron releases a voltage spike further when the membrane threshold value v_{thres} is exceeded, beyond which, the membrane value is reset to v_{reset} . After resetting this value, a neuron cannot fire for a limited time. So, the model neuron will have a potential value of membrane in resting state as E_{rest} and as E_{exc} and E_{inh} will be marked the equilibrium potential of excitatory/inhibitory synapses and

the conductances as g_{exc} and g_{inh} . Taking all these values, the voltage for a constant time τ can be defined as

$$\tau \frac{dV}{dt} = (E_{rest} - V) + g_{exc}(E_{exc} - V) + g_{inh}(E_{inh} - V). \quad (1)$$

2.2 Synapse

Synapses are a model of biological connections between neurons and operate on the principle of changing conductivity. Each synapse weights w (called synaptic weight), the value of which is increased when the presynaptic spike arrives. In the case when the spike does not come, the conductivity value decreases exponentially. In this model, if the neuron is stimulated then the conductance g_{exc} can be calculated as

$$\tau_{g_{exc}} \frac{dg_{exc}}{dt} = -g_{exc}. \quad (2)$$

In the same way, the value of $\tau_{g_{inh}}$ can be calculated in the following way

$$\tau_{g_{inh}} \frac{dg_{inh}}{dt} = -g_{inh}. \quad (3)$$

2.3 Spiking neural network and training algorithm

A spiking neural network can be built of two layers [9]. The first layer process the input which is built of $m \times n$ neurons due to the size of incoming data. Each neuron in the input layer has a Poisson firing (spike trains) [1]. The second layer has excitatory and inhibitory neurons. Each excitatory neuron is connected to one inhibitory neuron. In this model, if the excitatory neuron will send a spike, then only one inhibitory neuron receives it. But each inhibitory neuron is connected to almost all excitatory neurons (the exception is the neuron that sends a spike).

One of the known algorithms for training spiking neural networks is called spike-timing-dependent plasticity [22]. The idea is based on adding additional value x_{pre} to each synapse where the presynaptic trace is kept. This variable is used to count the number of arrived spikes through this synapse. This value is increased each time when there is a spike, but in many cases, there might not be. In this case, the value of x_{pre} is exponentially decreasing. The weights are calculated using a learning rate of η as

$$\Delta w = \eta(x_{tar} - x_{tar})(w_{max} - w)^\mu, \quad (4)$$

where x_{tar} is the value of output target, w_{max} is the maximum value of the weight, and a coefficient μ indicates a relationship to the previous weight.

3 Heuristic algorithms for hyper parameters

This section is focused on describing the heuristic idea of collaborative many populations for hyperparameter problem. The parameters that we take into account in the coefficients selection process are the number of neurons $n_{neurons}$, the value of E_{exc} , E_{inh} , μ , v_{thres} , v_{decay} (threshold potential decay). These values create a six-element vector that is identified with a single individual in the heuristic algorithm population.

3.1 General idea of heuristic approach

A heuristic is an algorithm that does not guarantee an ideal solution infinite time, so the obtained solution is an approximate one. In general, the heuristic algorithm is based on the selection of the initial population of individuals who are displaced in the solution space through the specified number of iterations. The movement takes place primarily as two operations, which can be interpreted as global and local. Global movement changes the actual position by a large number, and the local one changes only by a very small number. Both processes are important because global allows to avoid getting stuck in the local extreme, and local modifies the current value to increase accuracy relative to the ideal solution.

Each iteration in the heuristic algorithm is based on performing a global and local movement. Then, the individual is assessed for their fitness function and the best results can be treated as a solution. These operations are repeated until the stop condition is met which is the error value (when the solution is known), or the iteration number.

3.2 Collaborative work

Federate learning assumes that there will a few workers with their private dataset. Each of them will calculate some parameters and send them to one instance which will aggregate all of them. This process will be repeated for several iterations and in the end, the last aggregated results will be a configuration for some classifier. This idea can be transferred with some modification to heuristic operation with some modification. Let us assume that a set of workers will be marked as $\{\xi_1, \xi_2, \dots, \xi_{M-1}\}$, where $M > 0$ is the number of all workers and in our case corresponds to the number of processor threads. ξ_0 is treated as the main worker, which gets data from others and chose the best results. Each element, for instance ξ_j has a private dataset d_j .

Algorithm 1: The operation of worker ξ_0 in the proposed solution.

```

Input: The numbers of round  $K$  (where  $K \geq 1$ ), database  $D$ 
1 Split dataset  $D$  into  $M$  subsets;
2 Create a set of workers with a one subset  $d_i \in D$  and spiking neural network;
3  $t := 0$ ;
4 for  $t < T_{collaborative}$  do
5   for each worker  $i$  do
6     Wait for worker  $i$ ;
7     Retrain actual network by  $T_{train}$  iterations using local dataset  $d_0$ ;
8     Calculate the accuracy using local dataset  $d_0$ ;
9   Calculate the accuracy using averaged coefficients and local dataset  $d_0$ ;
10  Choose the best set of coefficients, by comparison, the accuracy levels;
11   $t++$ ;
12 Rerun the best set of coefficients;

```

Each worker with its dataset has one spiking neural network and one heuristic population composed of k individuals (as a six-elements vector). In the beginning, all individuals are randomly generated and for a specific number of iterations T_h , a heuristic algorithm is performed to find the best coefficients. Of course, the population must be evaluated in each iteration to avoid a random changing position. For that, we propose a fitness function that is based on calculating the accuracy of the trained network. To make it possible, the spiking neural network will be trained in each heuristic population for all individuals. To avoid huge amounts of calculations, the number of individuals k , as well as the number of training iteration T_{train} or heuristic iteration T_h on a dataset, should be adequate to the hardware capabilities. Each dataset d_j connected to a specific j worker (heuristic population) splits the dataset into two subsets – training $\Theta_{train(d_j)}$ and testing $\Theta_{val(d_j)}$. The second subset is used for validation in the fitness function. The main idea is to minimize loss function based on the mean squared error defined as

$$f(\Theta_{val(d_j)}) = \frac{1}{|\Theta_{val(d_j)}|} \sum_{i=1}^{|\Theta_{val(d_j)}|} (y_i - \hat{y}_i)^2, \quad (5)$$

where y_i is the true label of a tested sample x_i , and \hat{y}_i is the output from the network. However, there are other loss function like Contrastive Loss [25].

This means, that in each iteration on each worker, there will be T_h iteration. During one iteration of the heuristic approach, all individuals in the population are performing

the local and global movement. Then, all individuals are used to train spiking neural networks for T_{train} iterations on a private dataset of the worker. In the end, each individual is evaluated by Eq. (5). The individual with the lowest value of this function can be treated as a solution. Half the worst individuals are replaced by new random values and the best remains in the population.

The proposed idea is based on the actions of M such workers. It is worth noting that the database was divided into relation to all workers, so the ξ_0 has its database d_0 which is a subset of the whole database. However, this worker has no heuristics, only a spiking neural network. The main task of it is to wait for the best individual from the other workers. As a result, the ξ_0 will receive vectors with found coefficients. Based on these vectors, the worker chooses the best coefficients by finding the best using the following formulas as

$$\min\{f(\Theta_{val(d_0)})_{\xi_1}, f(\Theta_{val(d_0)})_{\xi_2}, \dots, f(\Theta_{val(d_0)})_{\xi_{M-1}}, f(\Theta_{val(d_0)})_{avg}\}, \quad (6)$$

where the last element uses the average coefficients from all workers what can be presented as

$$\left[\frac{1}{M-1} \sum_{i=1}^{M-1} c_0^{(i)}, \frac{1}{M-1} \sum_{i=1}^{M-1} c_1^{(i)}, \dots, \frac{1}{M-1} \sum_{i=1}^{M-1} c_6^{(i)} \right], \quad (7)$$

where each vector from worker i has the following form
,

where c_i means a specific coefficient.

Algorithm 2: The operation of worker ξ_i (where $i \in \{1, 2, M - 1\}$) in the proposed solution.

```

Input: Numbers of iteration  $T_h$ ,  $T_{train}$ 
1  $t = 0;$ 
2 for each worker  $i$  do
3   Create an initial population composed of  $k$  individuals; for each individual  $j$  do
4     Create a random set of variables in their range;
5 for  $t < T_h$  do
6   for each individual in population do
7     Perform a heuristic algorithm;
8     Train network with a dataset  $d_i$  by  $T_{train}$ ;
9   Calculate the accuracy for each individual;
10  if  $t < T_h$  then
11    The best 50% of them leave in population;
12    The worst 5% of the population replace with new values chosen in a random
        way;
13   $t ++;$ 
14 Return the best individual to worker  $\xi_0$ ;

```

When the best coefficient configuration is chosen by the worker ξ_0 , it is sent to all other workers. Each worker generates individuals in the population by giving them the obtained values with random changes at the level of $\langle -5\%, 5\% \rangle$ (only if the values are in a variable range). This operation is repeated until the specified amount of iterations $T_{collaborative}$ is reached. The whole process of the proposal is presented in Alg. 1–2 and an example visualization in Fig. 1.

4 Experiments

All experiments were made with the library BindsNET [12] which allows for simulation spiking neural networks and performed on a 12-thread Intel Core i7-8750H processor, 32GB RAM, and NVIDIA GeForce GTX 1050Ti Max-Q. A classification accuracy was analyzed on two datasets—MNIST [16] and Fashion MNIST [37]. Both datasets have 10 classes, 60000 training, and 10000 testing images.

Datasets were split with an equal division of samples according to classifying classes.

The analysis of the proposed approach was analyzed with 3, 6, and 12 workers. In the heuristic approach, we used 5 algorithms – CSA (Cuckoo Search Algorithm), WOA (Whale Optimization Algorithm), PBA (Polar Bears Algorithm), GOA (Grasshopper Optimization Algorithm), and SWA (Salp Swarm Algorithm). The parameters in these algorithms were set as – the size of population $k \in \{10, 25, 50, 100\}$, the number of iterations $T_h \in \{10, 25, 50, 100\}$, the rest of parameters depending on the model of the given algorithm were selected randomly. In the case of spiking neural networks, training iteration was set as $T_{train} \in \{400, 600, 800\}$. Moreover, the incoming sample to the network is a 2D image with size 28×28 , so the input layer will be composed of 784 neurons, where the incoming value of each pixel is presented in the form of a Poisson distribution with a firing ratio proportional to the pixel intensity in the database of analyzed images. As the maximum intensity

Fig. 1 Visualization of the idea of collaboration in the training process

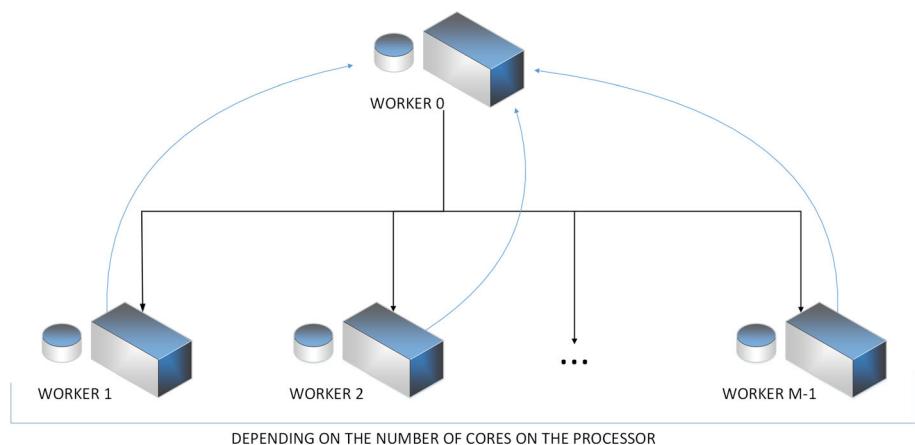


Table 1 Classification accuracy over the entire verification set after 3 iterations of collaboration using 3 workers

T_{train}	T_h	k	MNIST					FASHION MNIST				
			CSA	WOA	PBA	GOA	SWA	CSA	WOA	PBA	GOA	SWA
400	10	10	0,19	0,2	0,2	0,19	0,16	0,19	0,2	0,2	0,18	0,17
		25	0,1	0,1	0,2	0,1	0,18	0,09	0,1	0,22	0,19	0,18
		50	0,11	0,11	0,12	0,17	0,14	0,12	0,11	0,13	0,16	0,13
		100	0,18	0,18	0,15	0,18	0,16	0,18	0,19	0,16	0,18	0,17
	25	10	0,16	0,19	0,16	0,19	0,19	0,15	0,18	0,15	0,18	0,19
		25	0,22	0,16	0,2	0,15	0,19	0,22	0,17	0,22	0,15	0,19
		50	0,19	0,22	0,19	0,18	0,17	0,19	0,22	0,2	0,18	0,18
		100	0,17	0,16	0,18	0,16	0,18	0,18	0,15	0,19	0,17	0,17
	50	10	0,22	0,24	0,22	0,22	0,22	0,21	0,24	0,2	0,22	0,23
		25	0,23	0,24	0,24	0,24	0,25	0,24	0,22	0,26	0,24	0,26
		50	0,24	0,24	0,24	0,28	0,3	0,26	0,22	0,25	0,26	0,32
		100	0,31	0,3	0,32	0,31	0,31	0,33	0,27	0,33	0,34	0,28
	100	10	0,37	0,39	0,38	0,42	0,42	0,39	0,39	0,35	0,45	0,38
		25	0,44	0,39	0,41	0,39	0,44	0,48	0,4	0,39	0,38	0,42
		50	0,42	0,41	0,45	0,38	0,39	0,39	0,41	0,49	0,35	0,36
		100	0,42	0,43	0,46	0,43	0,42	0,38	0,45	0,46	0,43	0,46
600	10	10	0,19	0,13	0,18	0,12	0,16	0,2	0,13	0,19	0,12	0,16
		25	0,16	0,12	0,16	0,22	0,23	0,16	0,12	0,16	0,23	0,24
		50	0,13	0,12	0,17	0,13	0,13	0,14	0,11	0,18	0,13	0,14
		100	0,23	0,16	0,17	0,16	0,21	0,23	0,17	0,18	0,15	0,21
	25	10	0,22	0,21	0,22	0,21	0,22	0,22	0,23	0,21	0,21	0,22
		25	0,24	0,18	0,21	0,22	0,19	0,22	0,19	0,23	0,21	0,18
		50	0,2	0,2	0,22	0,22	0,21	0,2	0,18	0,21	0,21	0,23
		100	0,22	0,2	0,21	0,2	0,21	0,22	0,2	0,18	0,23	0,23
	50	10	0,26	0,24	0,21	0,24	0,25	0,28	0,22	0,19	0,22	0,27
		25	0,23	0,26	0,25	0,24	0,25	0,21	0,26	0,25	0,26	0,23
		50	0,31	0,27	0,27	0,32	0,31	0,33	0,26	0,27	0,33	0,29
		100	0,36	0,31	0,36	0,32	0,32	0,34	0,33	0,36	0,33	0,29
	100	10	0,45	0,38	0,44	0,43	0,45	0,48	0,38	0,42	0,41	0,48
		25	0,4	0,45	0,45	0,41	0,44	0,39	0,47	0,42	0,42	0,47
		50	0,45	0,46	0,43	0,42	0,46	0,45	0,44	0,43	0,44	0,45
		100	0,45	0,44	0,48	0,48	0,44	0,46	0,41	0,44	0,52	0,43
800	10	10	0,17	0,22	0,22	0,18	0,23	0,18	0,2	0,22	0,18	0,22
		25	0,16	0,19	0,22	0,14	0,15	0,16	0,18	0,23	0,15	0,15
		50	0,23	0,21	0,17	0,17	0,15	0,24	0,22	0,17	0,16	0,15
		100	0,19	0,14	0,17	0,21	0,15	0,19	0,13	0,17	0,22	0,15
	25	10	0,2	0,22	0,22	0,2	0,23	0,21	0,21	0,24	0,2	0,24
		25	0,22	0,2	0,21	0,23	0,24	0,23	0,21	0,23	0,22	0,24
		50	0,22	0,22	0,22	0,2	0,23	0,2	0,24	0,2	0,18	0,25
		100	0,23	0,22	0,21	0,21	0,22	0,21	0,21	0,19	0,2	0,2
	50	10	0,24	0,25	0,2	0,26	0,22	0,25	0,26	0,19	0,24	0,22
		25	0,25	0,23	0,25	0,24	0,25	0,26	0,23	0,27	0,23	0,23
		50	0,32	0,28	0,31	0,26	0,28	0,29	0,28	0,34	0,24	0,26
		100	0,34	0,33	0,3	0,35	0,36	0,31	0,36	0,28	0,32	0,39
	100	10	0,43	0,47	0,37	0,43	0,39	0,41	0,45	0,36	0,42	0,41
		25	0,39	0,42	0,47	0,38	0,4	0,4	0,43	0,49	0,36	0,37
		50	0,41	0,34	0,37	0,37	0,42	0,43	0,37	0,39	0,36	0,38
		100	0,49	0,4	0,4	0,39	0,46	0,45	0,37	0,37	0,37	0,43

Table 2 Classification accuracy over the entire verification set after 3 iterations of collaboration using 6 workers

T_{train}	T_h	k	MNIST				FASHION MNIST					
			CSA	WOA	PBA	GOA	SWA	CSA	WOA	PBA		
400	10	10	0,22	0,21	0,24	0,22	0,17	0,21	0,21	0,13	0,3	0,25
		25	0,11	0,13	0,22	0,11	0,2	0,2	0,12	0,2	0,01	0,1
		50	0,15	0,16	0,17	0,21	0,18	0,05	0,16	0,07	0,29	0,08
		100	0,21	0,25	0,21	0,22	0,22	0,21	0,35	0,29	0,2	0,32
	25	10	0,21	0,26	0,19	0,24	0,24	0,21	0,16	0,19	0,14	0,22
		25	0,27	0,23	0,22	0,16	0,23	0,16	0,21	0,31	0,16	0,12
		50	0,25	0,23	0,25	0,23	0,19	0,14	0,23	0,25	0,21	0,19
		100	0,22	0,18	0,24	0,22	0,19	0,11	0,28	0,32	0,32	0,18
	50	10	0,23	0,3	0,23	0,27	0,23	0,22	0,3	0,13	0,26	0,33
		25	0,25	0,25	0,3	0,3	0,28	0,23	0,32	0,19	0,2	0,3
		50	0,3	0,28	0,31	0,35	0,35	0,4	0,37	0,31	0,44	0,35
		100	0,35	0,37	0,38	0,32	0,37	0,33	0,26	0,37	0,43	0,27
	100	10	0,41	0,43	0,45	0,45	0,45	0,28	0,52	0,46	0,46	0,44
		25	0,51	0,45	0,43	0,43	0,51	0,51	0,41	0,41	0,3	0,51
		50	0,47	0,48	0,48	0,4	0,4	0,54	0,44	0,48	0,51	0,38
		100	0,53	0,56	0,6	0,54	0,56	0,54	0,6	0,6	0,5	0,63
600	10	10	0,21	0,15	0,2	0,16	0,21	0,21	0,24	0,3	0,06	0,1
		25	0,21	0,16	0,21	0,24	0,28	0,28	0,25	0,28	0,24	0,35
		50	0,16	0,13	0,18	0,15	0,18	0,25	0,22	0,08	0,05	0,27
		100	0,29	0,17	0,2	0,22	0,26	0,27	0,07	0,29	0,11	0,24
	25	10	0,29	0,24	0,27	0,22	0,29	0,19	0,13	0,25	0,22	0,38
		25	0,26	0,24	0,28	0,24	0,22	0,26	0,22	0,26	0,22	0,11
		50	0,21	0,25	0,29	0,27	0,28	0,11	0,23	0,37	0,38	0,27
		100	0,27	0,26	0,23	0,24	0,24	0,16	0,36	0,44	0,35	0,33
	50	10	0,29	0,29	0,23	0,28	0,32	0,37	0,29	0,3	0,39	0,22
		25	0,3	0,33	0,29	0,31	0,31	0,3	0,4	0,39	0,21	0,21
		50	0,37	0,32	0,33	0,38	0,35	0,44	0,32	0,21	0,26	0,24
		100	0,41	0,33	0,43	0,33	0,34	0,41	0,39	0,32	0,22	0,32
	100	10	0,49	0,39	0,5	0,49	0,48	0,5	0,39	0,46	0,58	0,37
		25	0,43	0,48	0,51	0,42	0,49	0,48	0,55	0,62	0,41	0,4
		50	0,52	0,47	0,5	0,49	0,47	0,48	0,44	0,48	0,57	0,48
		100	0,59	0,58	0,58	0,61	0,58	0,67	0,55	0,67	0,71	0,64
800	10	10	0,22	0,25	0,27	0,22	0,25	0,11	0,14	0,38	0,22	0,15
		25	0,2	0,21	0,24	0,17	0,19	0,3	0,28	0,35	0,07	0,19
		50	0,28	0,26	0,21	0,19	0,16	0,17	0,34	0,1	0,18	0,25
		100	0,24	0,15	0,21	0,23	0,19	0,23	0,25	0,21	0,22	0,19
	25	10	0,21	0,24	0,23	0,25	0,28	0,11	0,14	0,31	0,25	0,25
		25	0,29	0,23	0,28	0,29	0,29	0,38	0,32	0,17	0,36	0,18
		50	0,29	0,27	0,23	0,22	0,25	0,32	0,16	0,22	0,2	0,23
		100	0,26	0,26	0,22	0,27	0,25	0,33	0,25	0,33	0,44	0,49
	50	10	0,26	0,28	0,21	0,31	0,23	0,33	0,27	0,19	0,42	0,32
		25	0,28	0,3	0,29	0,26	0,31	0,28	0,29	0,27	0,26	0,31
		50	0,33	0,33	0,37	0,27	0,31	0,33	0,21	0,46	0,17	0,38
		100	0,41	0,36	0,37	0,37	0,37	0,47	0,25	0,26	0,43	0,36
	100	10	0,44	0,51	0,4	0,49	0,46	0,52	0,56	0,36	0,6	0,42
		25	0,41	0,47	0,48	0,43	0,42	0,41	0,35	0,35	0,4	0,3
		50	0,53	0,46	0,42	0,45	0,5	0,52	0,54	0,43	0,46	0,48
		100	0,66	0,65	0,65	0,59	0,7	0,64	0,61	0,59	0,66	0,76

Table 3 Classification accuracy over the entire verification set after 3 iterations of collaboration using 12 workers

T_{train}	T_h	k	MNIST				FASHION MNIST			
			CSA	WOA	PBA	GOA	SWA	CSA	WOA	PBA
400	10	10	0.32	0.51	0.24	0.62	0.47	0.3	0.33	0.23
		25	0.21	0.23	0.22	0.11	0.4	0.2	0.21	0.3
		50	0.25	0.16	0.07	0.51	0.38	0.23	0.4	0.33
		100	0.51	0.55	0.61	0.42	0.32	0.48	0.5	0.56
	25	10	0.51	0.16	0.29	0.14	0.34	0.47	0.3	0.37
		25	0.27	0.33	0.52	0.16	0.43	0.25	0.3	0.33
		50	0.35	0.53	0.25	0.33	0.39	0.33	0.48	0.46
		100	0.42	0.38	0.54	0.42	0.39	0.38	0.36	0.35
	50	10	0.43	0.4	0.33	0.47	0.33	0.39	0.37	0.43
		25	0.35	0.45	0.3	0.5	0.18	0.32	0.41	0.46
		50	0.4	0.58	0.51	0.45	0.45	0.36	0.55	0.43
		100	0.45	0.57	0.58	0.42	0.37	0.42	0.53	0.38
	100	10	0.31	0.63	0.45	0.75	0.65	0.29	0.59	0.43
		25	0.61	0.75	0.43	0.53	0.71	0.57	0.7	0.49
		50	0.57	0.68	0.68	0.8	0.4	0.54	0.62	0.74
		100	0.63	0.96	0.7	0.74	0.9	0.59	0.87	0.67
600	10	10	0.41	0.25	0.3	0.26	0.11	0.39	0.24	0.28
		25	0.31	0.46	0.31	0.44	0.68	0.28	0.33	0.29
		50	0.26	0.43	0.08	0.25	0.38	0.24	0.4	0.08
		100	0.49	0.07	0.5	0.22	0.56	0.45	0.2	0.47
	25	10	0.49	0.44	0.47	0.22	0.39	0.46	0.41	0.45
		25	0.26	0.34	0.48	0.34	0.32	0.24	0.31	0.46
		50	0.21	0.45	0.49	0.47	0.28	0.2	0.42	0.45
		100	0.37	0.66	0.13	0.64	0.64	0.35	0.6	0.58
	50	10	0.69	0.29	0.23	0.68	0.22	0.66	0.27	0.21
		25	0.6	0.73	0.69	0.21	0.31	0.55	0.7	0.63
		50	0.57	0.62	0.33	0.38	0.55	0.54	0.58	0.31
		100	0.61	0.53	0.43	0.33	0.54	0.58	0.5	0.3
	100	10	0.59	0.49	0.8	0.59	0.68	0.55	0.47	0.54
		25	0.53	0.78	0.91	0.52	0.49	0.49	0.71	0.84
		50	0.82	0.67	0.5	0.79	0.67	0.76	0.64	0.46
		100	0.69	0.58	0.78	0.71	0.91	0.66	0.55	0.73
800	10	10	0.42	0.45	0.57	0.42	0.45	0.38	0.42	0.52
		25	0.3	0.51	0.44	0.17	0.39	0.28	0.39	0.33
		50	0.38	0.36	0.41	0.49	0.56	0.35	0.34	0.39
		100	0.34	0.35	0.31	0.33	0.19	0.32	0.32	0.29
	25	10	0.41	0.34	0.63	0.25	0.18	0.37	0.31	0.58
		25	0.49	0.63	0.18	0.69	0.19	0.47	0.59	0.17
		50	0.49	0.37	0.43	0.32	0.25	0.45	0.34	0.39
		100	0.36	0.56	0.32	0.57	0.45	0.34	0.52	0.42
	50	10	0.36	0.28	0.31	0.51	0.43	0.34	0.26	0.29
		25	0.48	0.5	0.29	0.46	0.31	0.45	0.46	0.27
		50	0.53	0.33	0.57	0.37	0.71	0.5	0.3	0.53
		100	0.51	0.26	0.27	0.57	0.57	0.47	0.24	0.25
	100	10	0.74	0.61	0.4	0.89	0.56	0.69	0.56	0.37
		25	0.51	0.57	0.68	0.43	0.42	0.48	0.53	0.62
		50	0.73	0.86	0.52	0.45	0.5	0.7	0.8	0.48
		100	0.89	0.75	0.89	0.69	0.8	0.81	0.72	0.82

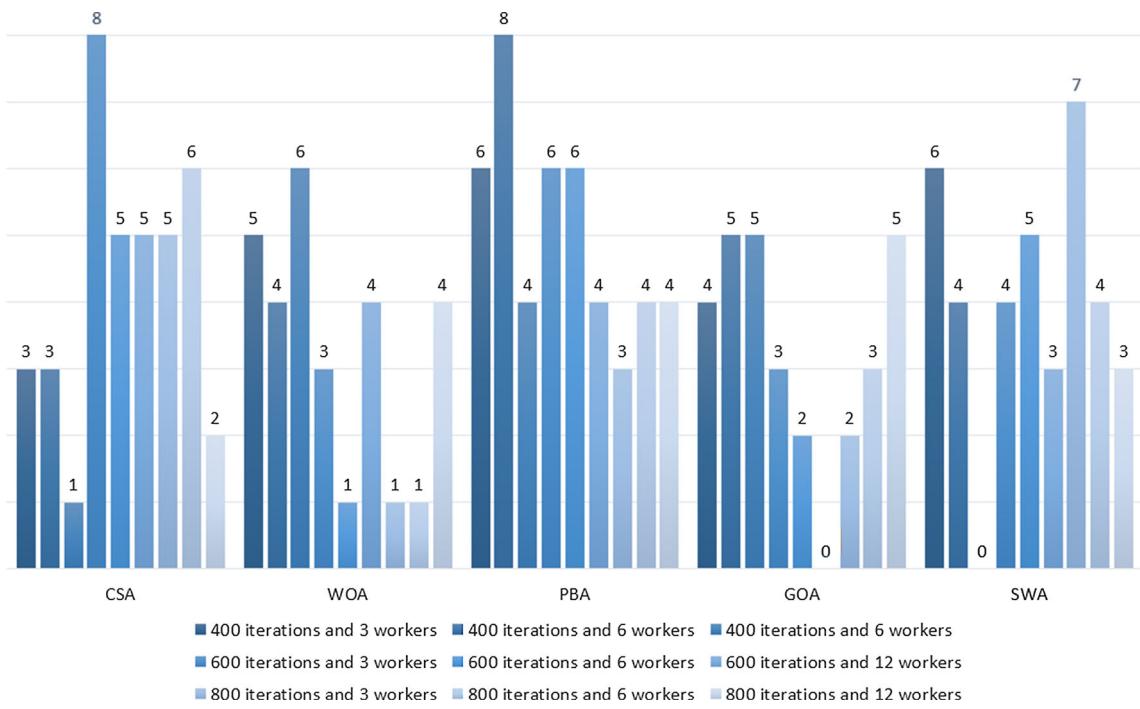


Fig. 2 Ranking of heuristics that obtained the highest accuracy in the classification problem of MNIST database

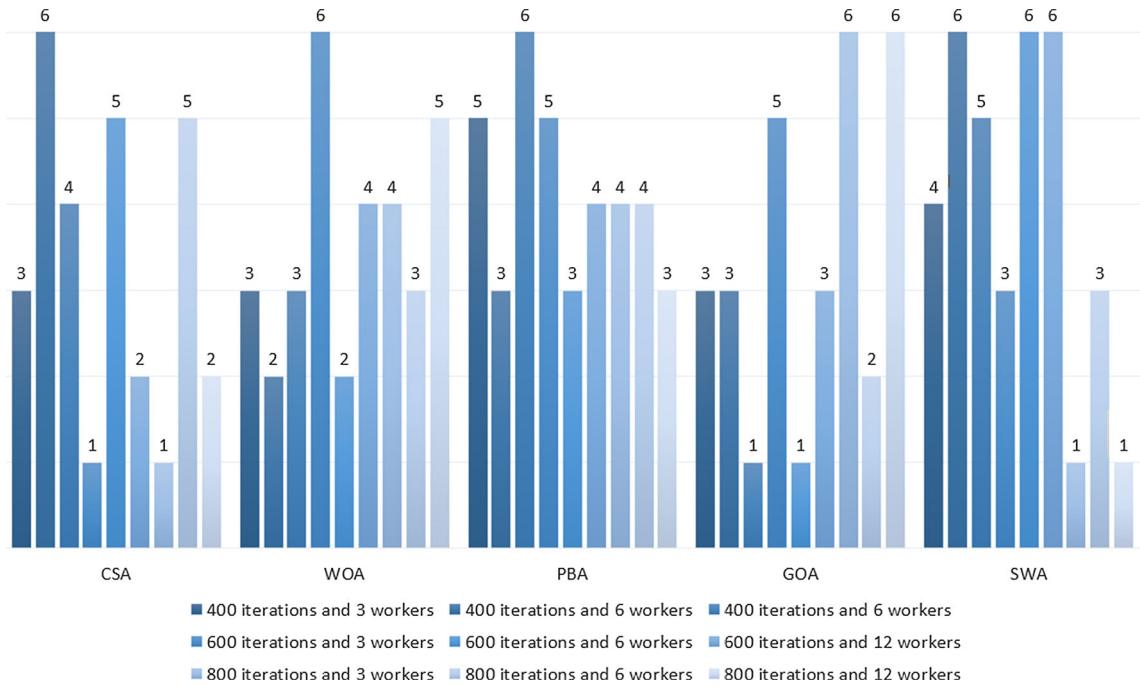


Fig. 3 Ranking of heuristics that obtained the highest accuracy in the classification problem of Fashion MNIST database

value (through the RGB model) the value of $255/4$ is chosen, i.e. 66.75 Hz, so it gives the value of fire from the range $(0, 66.75)$ Hz.

In all conducted experiments, we measured the final accuracy and time. In the case of the accuracy, created a vector with coefficients' values were used to train spiking

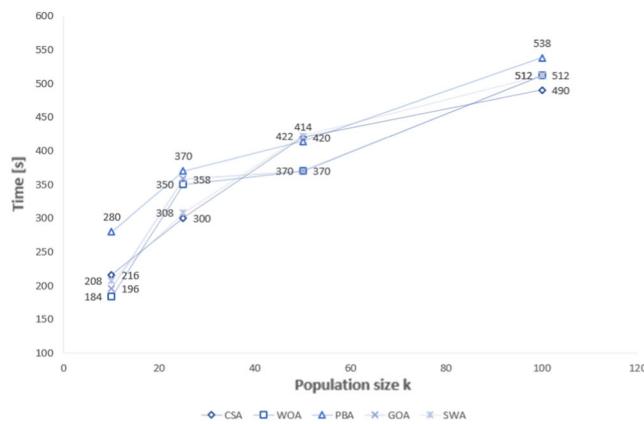
neural network (with the same architecture) by 800 iterations on the whole training dataset. The validation set was used for calculating the final accuracy for both classification problems. The results for the MNIST and Fashion MNIST databases are presented in Tables 1, 2 and 3 (respectively, for 3, 6 and 12 workers). In both cases, using

only 3 workers despite the chosen heuristic gives very low accuracy except the situation with 12 workers. These results can be justified through much greater coefficient analysis. For using only three workers, the number of coefficient configurations analyzed is four times smaller than in the higher number of workers. In most heuristics, the classifier accuracy was close to 50% which is very high classification results taking into account a small number of iterations of heuristics. We notice that in some algorithms like CSA, PBA, and GOA, the population stuck in the local extremum and had problems with escaping. This situation often resulted in a worse accuracy level than other algorithms. However, this is not a rule because, for 12 workers and PBA, the accuracy of the classifier was the highest. The accuracy level increased with the number not only of the workers but also of training iterations as well as the iterations of individual movements in the heuristic algorithm. An increase in any of the parameters allows for increasing the quality of the proposed technique.

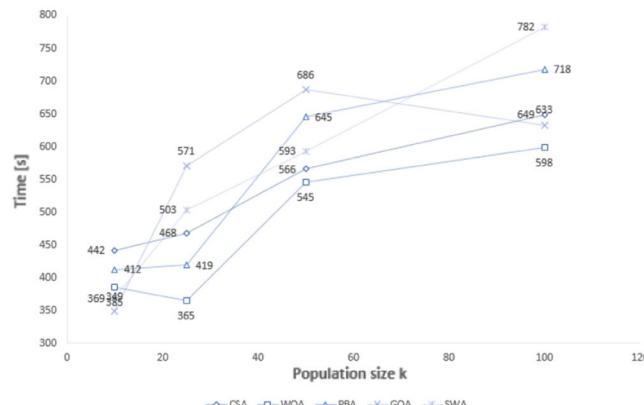
We counted the best results in each test and presented the results in Figs. 2 and 3 to visualize the heuristics algorithm selection. In the case of the MNIST database,

PBA, CSA, and SWA algorithms achieved the best results in terms of quantity. Two algorithms, i.e. CSA and PBA, showed the best results using a specific number of workers—CSA for 3, and PBA for 6. For the second database, the best results were achieved using PBA, SWA, and WOA. It is worth noting that PBA and SWA achieved good results with the other algorithms considering stability. By stability, we mean good results regardless of the workers or iteration number. In general, the best accuracy level was achieved by different coefficients value, where only the number of neurons was in the range of $\langle 100, 150 \rangle$.

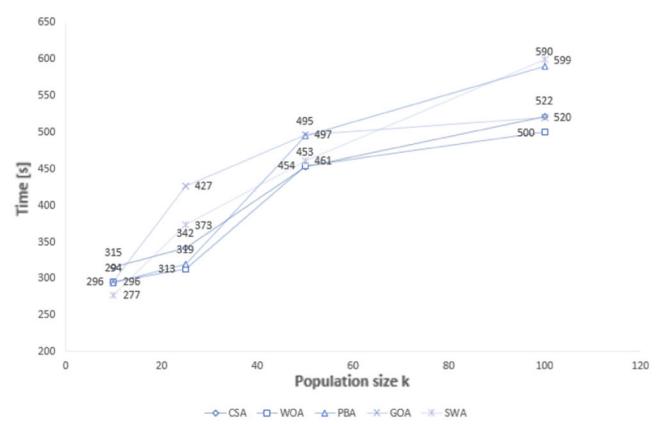
During the experiments, we analyzed also the time needed to find the best coefficients used in heuristic algorithms. For each set of such coefficients, the average time needed to perform calculations for one worker during one heuristic search. It should be noted that the more workers, the private database are smaller (due to equal distribution). Also, for a small number of individuals, the average time is indirect taking into account the measurements with the minimum and the maximum number of training iterations. Obtained results are presented in relation to different numbers of $T_{\text{heuristics}}$ and the number of workers in Figs. 4,



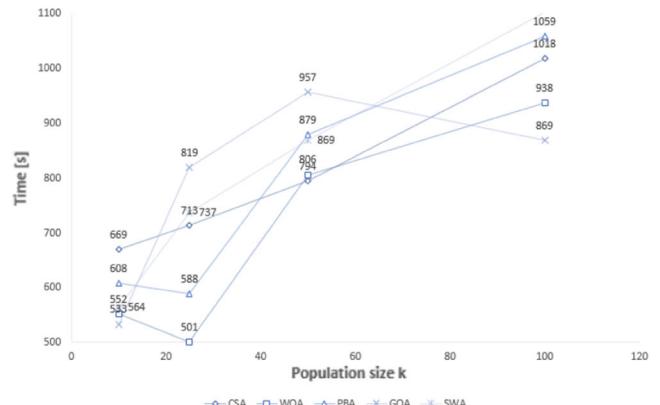
(a) 10 iterations.



(c) 50 iterations.



(b) 30 iterations.



(d) 100 iterations.

Fig. 4 Averaged time for both datasets using 3 workers

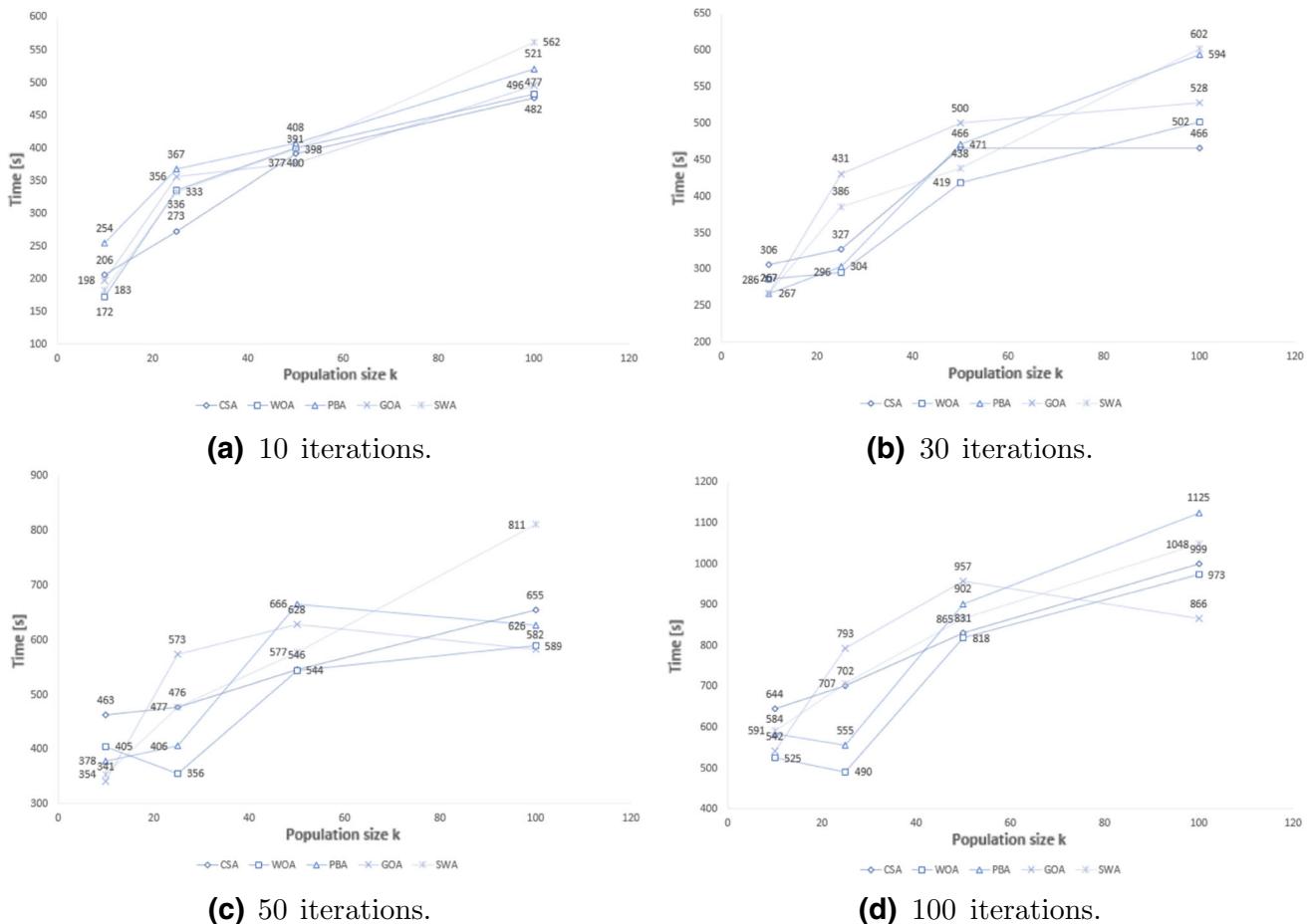


Fig. 5 Averaged time for both datasets using 6 workers

5 and 6. On these charts, the average time needed to perform each of the selected heuristic algorithms takes into consideration three parameters – the specific number of iterations, number of individuals in the population, and the number of a worker in federated learning. Based on all of these charts, the time increases with a higher value of any parameter. It is caused by the larger amount of calculations. The most optimal solution is the use of 50 iterations, which obtained an average time of about 500–600 seconds for one worker which in one iteration reaches an average accuracy of 50% (see Figs. 4c, 5c and 6c).

The highest accuracy was obtained using the maximum values of the analyzed parameters (the number of iterations and population size), although the time needed to accomplish this task for one worker increases by nearly 50–60% relative to the smaller number of iterations (Fig. 6d). All performed experiments show that the best results are gained using a large number of basic coefficients in heuristic algorithms (like the number of iteration and population size). It is caused by analyzing the bigger number of possibilities by the heuristic algorithm. However, it causes that time needed for it is very high. To avoid

such a long calculation time, it is possible to use a less time-consuming variant (smaller number of iterations/individuals) and longer retraining of the final classifier with the obtained parameters after using specific values of coefficients in the heuristic, the final retraining can be performed with a larger number of training iteration. Based on time measurements, most algorithms showed a linear increase relative to the number of individuals in the population (what can be noticed in the example of CSA). Only in a few cases such as the GOA algorithm, the situation was different, what can be seen in Figs. 4 and 5. In this case, the average time curve increases to 50 individuals, and at 100 it is already much lower, for instance with using 50 or 100 iterations on 3 workers. We also compared the GOA with a classic approach which is the Bayesian approach (BA) [3]. This set of parameters examined by BA reached 83% for the MNIST database and 77% for MNIST Fashion. These results show a high accuracy metric, but the proposal reached, respectively, 89%, and 82%. This 5–6% of accuracy was obtained by the use of the highest values in the conducted tests.

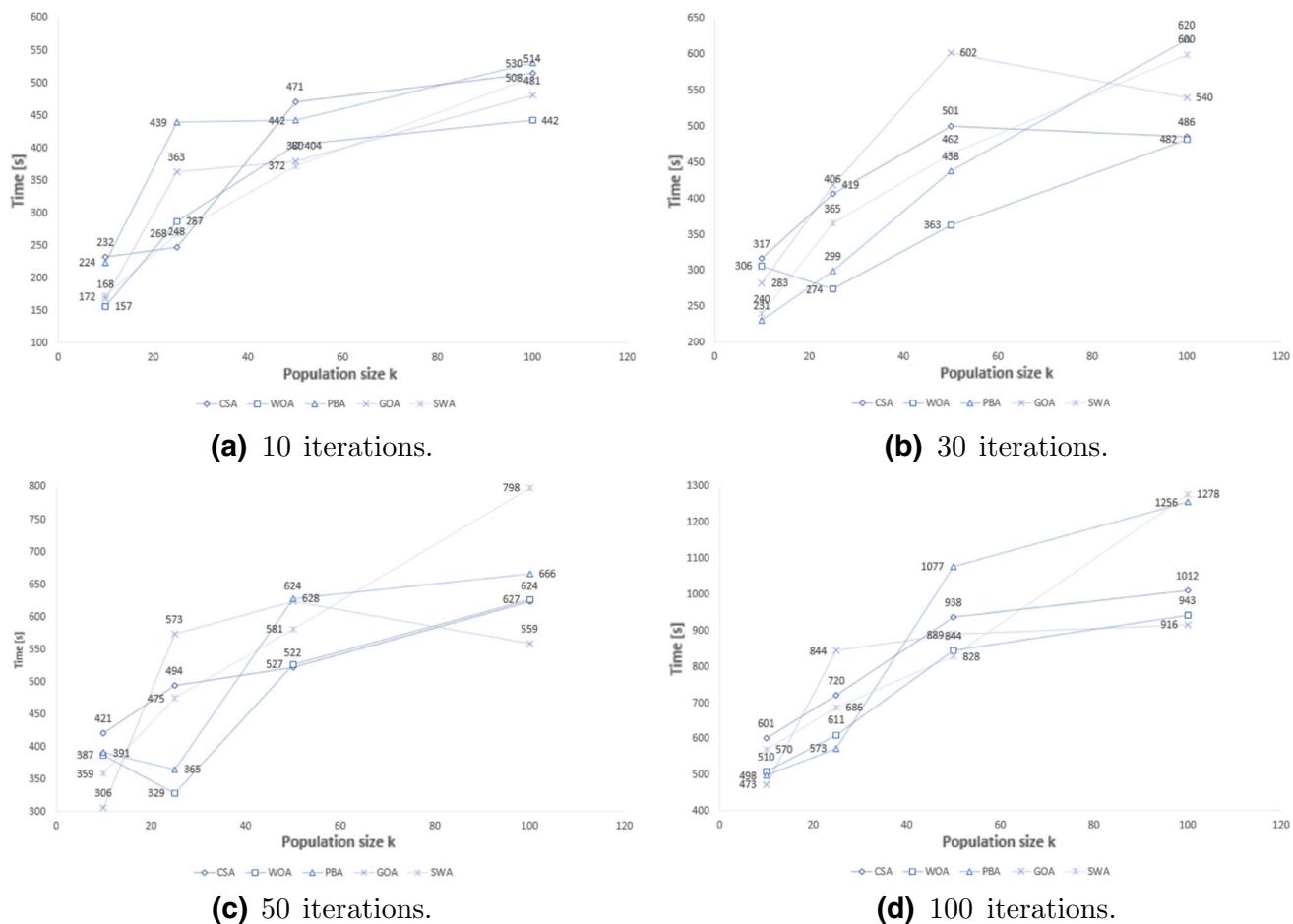


Fig. 6 Averaged time for both datasets using 12 workers

5 Conclusion

The selection of parameters in the training process not only neural networks but other artificial intelligence methods is an important and open problem. In this paper, we presented the idea based on parallel collaborative learning to find network parameters in such a way that the obtained accuracy is as high as possible. The main idea was to use federated learning to find the best parameter values. In conducted experiments, we analyzed three scenarios of federated learning with the use of 3, 6, and 12 workers. Moreover, each worker performs one of five different heuristic algorithms inspired by nature. In each of them, we examined the impact of the population size and iterations. Based on this setup, the best results were reached by CSA, PBA, and SWA for a small number of iterations and workers. In this case, when the number of workers was 12, the best results were reached by GOA, PBA, and WOA. In all cases, when the number of these parameters was increased, the time needed to calculate it was also higher. The more workers in federated learning are used, the higher level of accuracy can be achieved faster. The tests

showed that the developed technique can be used in practice for this purpose. However, the best results were obtained using many threads. During the research, a large number of calculations were also noted, which is reduced by using multi-threading and dividing the training database into equal parts for each thread. Hence, not only do the heuristic coefficients have a large impact, but also the number of workers.

In future research, we plan to focus on reducing the number of calculations, and thus also the training time. This could be done by the use of parallel operations as well as threads, where the worker could finish calculations faster. An additional element that should be analyzed is the number of training iterations.

Declaration

Conflict of interest The authors have no conflict of interest to declare.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing,

adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Averbeck BB (2009) Poisson or not poisson: differences in spike train statistics between parietal cortical areas. *Neuron* 62(3):310–311
2. Balaji A, Das A, Wu Y, Huynh K, Dell’Anna FG, Indiveri G, Krichmar JL, Dutt ND, Schaafsma S, Catthoor F (2019) Mapping spiking neural networks to neuromorphic hardware. *IEEE Trans Very Large Scale Integr (VLSI) Syst* 28(1):76–86
3. Balandat M, Karrer B, Jiang DR, Daulton S, Letham B, Wilson AG, Bakshy E (2020) Botorch: A framework for efficient monte-carlo bayesian optimization. In: NeurIPS
4. Bernert M, Yvert B (2019) An attention-based spiking neural network for unsupervised spike-sorting. *Int J Neural Syst* 29(08):1850,059
5. Capecci E, Lobo JL, Laña I, Espinosa-Ramos JI, Kasabov N (2019) Modelling gene interaction networks from time-series gene expression data using evolving spiking neural networks. *Evolv Syst* 1–15
6. Cheng L, Liu Y, Hou ZG, Tan M, Du D, Fei M (2019) A rapid spiking neural network approach with an application on hand gesture recognition. *IEEE Trans Cognit Develop Syst* 13(1):151–161
7. Czerpak P, Artiemjew P (2012) Job scheduling algorithm based on multi criteria optimization. In: Studies & Proceedings of Polish Association for Knowledge Management, 60
8. Dąbrowska D, Kucharski R, Witkowski AJ (2016) The representativity index of a simple monitoring network with regular theoretical shapes and its practical application for the existing groundwater monitoring network of the tychy-urbanowice landfills, Poland. *Environ Earth Sci* 75(9):749
9. Diehl PU, Cook M (2015) Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front Comput Neurosci* 9:99
10. Donati E, Payvand M, Risi N, Krause R, Indiveri G (2019) Discrimination of EMG signals using a neuromorphic implementation of a spiking neural network. *IEEE Trans Biomed Circuits Syst* 13(5):795–803
11. Haessig G, Berthelon X, Ieng SH, Benosman R (2019) A spiking neural network model of depth from defocus for event-based neuromorphic vision. *Sci Rep* 9(1):1–11
12. Hazan H, Saunders DJ, Khan H, Patel D, Sanghavi DT, Siegelmann HT, Kozma R (2018) Bindsnet: a machine learning-oriented spiking neural networks library in python. *Front Neuroinf* 12:89. <https://doi.org/10.3389/fninf.2018.00089>
13. Hyla T, Wawrzyniak N (2021) Identification of vessels on inland waters using low-quality video streams. In: Proceedings of the 54th Hawaii International Conference on System Sciences, p 7269
14. Kristensen L, Degroote M, Wittek P, Aspuru-Guzik A, Zinner N (2020) Artificial spiking quantum neural networks. *Bull Am Phys Soc*
15. Kumarasinghe K, Kasabov N, Taylor D (2020) Deep learning and deep knowledge representation in spiking neural networks for brain-computer interfaces. *Neural Netw* 121:169–185
16. LeCun Y, Cortes C, Burges C (2010) Mnist handwritten digit database
17. Lee C, Sarwar SS, Panda P, Srinivasan G, Roy K (2020) Enabling spike-based backpropagation for training deep neural network architectures. *Front Neurosci* 14
18. Li L, Jamieson K, DeSalvo G, Rostamizadeh A, Talwalkar A (2017) Hyperband: a novel bandit-based approach to hyperparameter optimization. *J Mach Learn Res* 18(1):6765–6816
19. Lin JCW, Ahmed U, Srivastava G, Wu JMT, Hong TP, Djenouri Y (2021) Linguistic frequent pattern mining using a compressed structure. *Appl Intell*, 1–18
20. Mardani A, Liao H, Nilashi M, Alrasheedi M, Cavallaro F (2020) A multi-stage method to predict carbon dioxide emissions using dimensionality reduction, clustering, and machine learning techniques. *J Cleaner Prod* 275:122,942
21. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv. Eng. Softw.* 95:51–67
22. Morrison A, Aertsen A, Diesmann M (2007) Spike-timing-dependent plasticity in balanced random networks. *Neural Comput* 19(6):1437–1467
23. Nour M, Öztürk Ş, Polat K (2021) A novel classification framework using multiple bandwidth method with optimized CNN for brain-computer interfaces with EEG-fNIRS signals. *Neural Comput Appl* 33(22):15815–15829
24. Okuyama Y, Abdallah AB et al (2019) Analytical performance assessment and high-throughput low-latency spike routing algorithm for spiking neural network systems. *J Supercomput* 75(8):5367–5397
25. Öztürk Ş (2021) Comparison of pairwise similarity distance methods for effective hashing. In: IOP Conference Series: Materials Science and Engineering, vol. 1099, p. 012072. IOP Publishing
26. O’Connor P, Gavves E, Welling M (2019) Training a spiking neural network with equilibrium propagation. In: The 22nd international conference on artificial intelligence and statistics, pp 1516–1523
27. Połap D et al (2017) Polar bear optimization algorithm: meta-heuristic with fast population movement and dynamic birth and death mechanism. *Symmetry* 9(10):203
28. Rongala UB, Mazzoni A, Spanne A, Jörnstell H, Oddo CM (2020) Cuneate spiking neural network learning to classify naturalistic texture stimuli under varying sensing conditions. *Neural Netw* 123:273–287
29. Saremi S, Mirjalili S, Mirjalili S, Dong JS (2020) Grasshopper optimization algorithm: theory, literature review, and application in hand posture estimation. In: Nature-Inspired Optimizers, pp 107–122. Springer
30. Saunders DJ, Patel D, Hazan H, Siegelmann HT, Kozma R (2019) Locally connected spiking neural networks for unsupervised feature learning. *Neural Netw* 119:332–340
31. Sengupta A, Ye Y, Wang R, Liu C, Roy K (2019) Going deeper in spiking neural networks: VGG and residual architectures. *Front Neurosci* 13
32. Tavanaei A, Ghodrati M, Kheradpisheh SR, Masquelier T, Maida A (2019) Deep learning in spiking neural networks. *Neural Netw* 111:47–63
33. Tubishat M, Idris N, Shuib L, Abushariah MA, Mirjalili S (2020) Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Syst Appl* 145:113,122
34. Wajda A, Jaworski T (2021) Optimization and security of hazardous waste incineration plants with the use of a heuristic algorithm. *Sensors* 21(21):7247

35. Wang W, Hao S, Wei Y, Xiao S, Feng J, Sebe N (2019) Temporal spiking recurrent neural network for action recognition. *IEEE Access* 7:117,165–117,175
36. Włodarczyk-Sielicka M, Stateczny A (2017) Fragmentation of hydrographic big data into subsets during reduction process. In: 2017 Baltic Geodetic Congress (BGC Geomatics), pp 193–198. IEEE
37. Xiao H, Rasul K, Vollgraf R (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint [arXiv:1708.07747](https://arxiv.org/abs/1708.07747)
38. Yang L, Shami A (2020) On hyperparameter optimization of machine learning algorithms: theory and practice. *Neurocomputing* 415:295–316
39. Yang XS, Deb S (2009) Cuckoo search via lévy flights. In: 2009 World congress on nature & biologically inspired computing (NaBIC), pp 210–214. IEEE

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.