

Research Article

Neural Nets Distributed on Microcontrollers using Metaheuristic Parallel Optimization Algorithm

Fazal Noor* and Hatem ElBoghdadi

Faculty of Computer and Information Systems, Islamic University of Madinah, Saudi Arabia

mfnoor@iu.edu.sa; helboghdadi@iu.edu.sa

*Correspondence: mfnoor@gmail.com

Received: 14th April 2020; Accepted: 29th August 2020; Published: 1st October 2020

Abstract: Metaheuristic algorithms are powerful methods for solving compute intensive problems. neural Networks, when trained well, are great at prediction and classification type of problems. Backpropagation is the most popular method utilized to obtain the weights of Neural Nets though it has some limitations of slow convergence and getting stuck in a local minimum. In order to overcome these limitations, in this paper, a hybrid method combining the parallel distributed bat algorithm with backpropagation is proposed to compute the weights of the Neural Nets. The aim is to use the hybrid method in applications of a distributed nature. Our study uses the Matlab[®] software and Arduino[®] microcontrollers as a testbed. To test the performance of the testbed, an application in the area of speech recognition is carried out. Due to the resource limitations of Arduino microcontrollers, the core speech pre-processing of LPC (linear predictive coding) feature extractions are done in Matlab[®] and only the LPC parameters are passed to the Neural Nets, which are implemented on Arduino microcontrollers. The experimental results show that the proposed scheme does produce promising results.

Keywords: *Bat algorithm; Genetic algorithm; Neural Networks; Speech recognition; Optimization algorithm*

1. Introduction

Metaheuristic optimization algorithms have been used in many areas such as biology, chemistry, control systems, engineering and many other fields which require huge amounts of computations. The Bat algorithm (BA) is a metaheuristic optimization algorithm that has been reported to be efficient in providing optimal solutions to continuous nonlinear constrained problems [1, 2].

Artificial neural networks have been applied in many fields such as smart devices, automotive industry, enterprise organizations, medicine, drug discovery and many others [3]. A neural network is adaptive in nature such that the internal structure has layers of neurons with activation functions. These neurons are connected with other neurons by links. The weights of the links are adjusted iteratively by calculating an error function at the output. Once the desired output behaviour is achieved, the neural network is said to have been trained and prepared to be tested on the new unseen input data. Neural nets often use backpropagation, a popular method to obtain the weights of the links in the hidden inner neural nets. However, using backpropagation has been shown to be slow in convergence and often converges to a local minimum [4, 5].

Even though neural networks and metaheuristic optimization algorithms have been applied in numerous research fields, the paper presents two contributions, first a hybrid method is proposed to overcome the problem of convergence to a local minimum and usage of the parallel distributed bat algorithm in parallel neural networks [6]. Note, the BAT-BP proposed by Nazri [23] does not use

parallel methods. The second is to implement them on low cost microcontrollers which usually have limited onboard hardware resources [7]. There are many applications where such an architecture may be utilized, one application consisting of speech recognition is presented.

The paper is organized as follows: Section Two presents the literature review, Section Three presents the parallel distributed bat algorithm, Section Four presents the background on neural networks. Section Five presents the results and finally, Section Six provides the conclusion and future work.

2. Literature Review

Many metaheuristic optimization algorithms were devised by imitating nature [1,13,14]. These algorithms are first initialized with a random population as an input. The process is iterative in nature and at every iteration a fitness function is computed and the values obtained are used for guidance to an optimum solution. The obtained solution does not have to be exact, however, approximately close such that it gives a satisfactory result. Neural Networks, on the other hand, are inherently parallel in nature and have been utilized in numerous applications [3].

Neural Networks are mainly used for prediction or classification type of problems [8-12]. Incorporating the heuristic optimization algorithms with neural networks were done long time ago. However, with advancement in computational power of processors and reduction in cost of equipment, more research is being done with applications in numerous areas, to mention a few are cancer cell detection, speech recognition and drug side effect prediction and/or classification. These are application areas demanding intensive computations. With open source microcontrollers available on the market, it becomes interesting to implement Neural networks on them and study their performance.

In deep learning, the model learns to perform classification directly, for example, from sound or images. Deep learning (see Fig. 1) is a neural network architecture with many hidden layers which can run in hundreds of layers. Some examples of deep learning are the following: an autonomous vehicle stops or slows down at a pedestrian crosswalk, as a counterfeit bank note is put in an ATM machine it calls the security and the culprit gets apprehended, another example is a smartphone application that provides translation of a foreign street sign instantly. Deep learning is well suited for identification type of problems such as face recognition, speech recognition, sign recognition, and many others.

Object recognition have been possible with fairly high accuracy. There are data sets such as ImageNet which are available for free and can be used to train the neural networks. For example, a neural network trained on images can identify which type of vehicle it is, i.e. a car, truck, or bicycle. With increased computing power it is possible to accelerate the training of huge amounts of data and reduce the training time from months or weeks, to days or hours.

AlexNet a popular net available on the internet is being used to perform new recognition tasks and was trained with 1.3 million images of high resolution which is capable of recognizing 1000 different objects. A deep neural network can be developed for nonlinear processing using simple elements or processors operating in parallel. The human brain is similar to it. The deep neural work consists of an input layer, many hidden layers and an output layer. The layers are interconnected via nodes (called neurons). The network learns from the inputs, we do not have any idea about the features, the network learns by itself.

Convolutional neural networks (CNN) are one of the most well-known algorithms for deep learning for images and videos. The input is first convolved, then pooled and then sent through a rectified linear unit. The three operations are repeated over and over again with tens or hundreds of layers and each layer learns new and different features. After feature detection, the second phase is the classification. The final layer is using a softmax function for the classification output as shown in Fig. 1 below.

There is a difference between deep learning and machine learning. Deep learning is a type of machine learning. In machine learning the features are manually extracted for example from an image. Whereas with deep learning, the images are fed into a neural network and features are learned

automatically. For deep learning, the learning phase may require hundreds of thousands or up to millions of images for better results. Since this phase is computationally intensive it is better to have extremely fast CPUs, or GPUs (Graphical Processing Units) or PC clusters. Table 1 below, shows the relations between machine learning and deep learning.

Table 1. Comparison between machine learning and deep learning.

Machine Learning	Deep Learning
Good results are achieved with small data sets	Demands very large data sets
Quick to train a model	Computationally demanding
Different features and classifiers are required to achieve best or optimum results	Automatically learns features and classifiers
Accuracy levels off (saturates)	Accuracy is high

Deep learning requires intensive computations and therefore the selection of resources is critical as it can take hours, weeks, months or even years to obtain accurate results due to huge amounts of data and computational power. For computation there are several options, CPU based, GPU based, PC cluster based and cloud based. CPU based is best for simple pretrained Neural Networks and is a readily available option. A GPU can reduce network training time from weeks to days or days to hours depending on the data and problem at hand. Multiple GPUs are often used to further speed up the processing time. A PC cluster-based solution is a good option as if one has access to a supercomputer at an affordable price [15]. It is useful in applications which can be run in parallel and demands less communication time compared to computation time [15-20]. The Cloud based platform is best as there is no purchase of any equipment nor setup or configuration of the hardware. One example of a practical study is in Cancer diagnostics with Deep Learning and Photonic time stretch [22]. Cancer patients usually receive chemotherapy and must on a regular basis undergo CT (computerized tomography) and PET (positron emission tomography) scans to check on the progress of the treatment. A method called Flow cytometry is used to identify tumor cells flowing in the blood and is considered a critical parameter in cancer treatment [24]. However, in this method extremely high volumes of data is generated, in fact 100 gigabytes of data “per second. For a single experiment, in which every cell in a 10-milliliter blood sample is imaged at almost 100,000 cells per second, the system generates from 10 to 50 terabytes of data” [24]. As it involves big data, it takes more than a week to complete the image processing and machine learning processes. Parallel computing with 16 processors is used to accelerate and reduce the time required to complete the analysis from a week to half a day [24]. Similarly, speech recognition requires many samples of recorded voice to train a neural network [21].

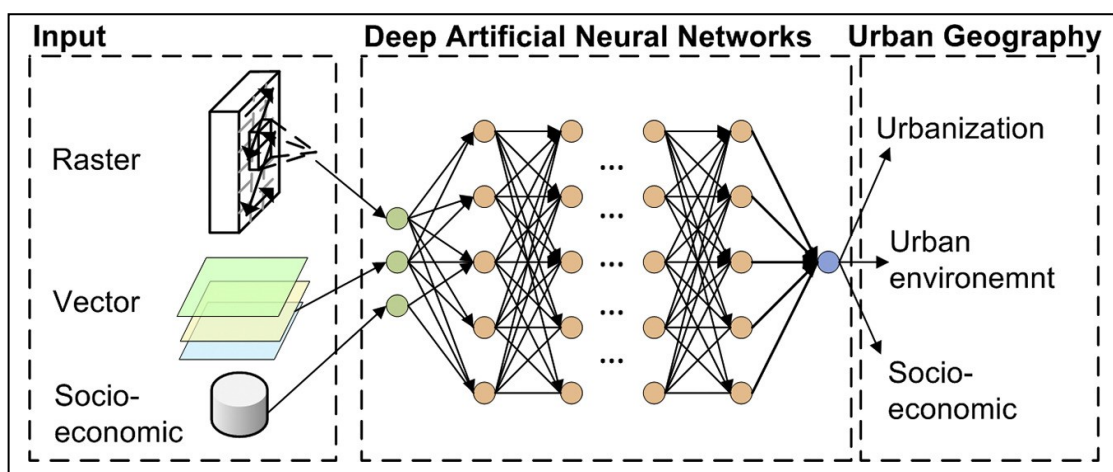


Figure 1. Deep Learning Neural Network¹ [25].

Although previous work in the literature appears to use a mixture of optimization methods replacing the backpropagation method for the weight calculations. Our proposed method is novel in that it is using parallel distributed bat algorithm with backpropagation method in parallel neural

¹ https://ars.els-cdn.com/content/image/1-s2.0-S0198971518302928-ga1_lrg.jpg

networks. To the best of our knowledge, our proposed method does not appear in the literature at the time of our writing of this paper.

3. Parallel Distributed Optimization Method using Bat Algorithm
Xin-She Yang in 2010, developed a new meta-heuristic optimization algorithm based on the echolocation behaviour of a bat [1]. The bats transmit pulses of sound waves and listen to the reflected echoes. From the reflected waves, the bats are able to differentiate between prey and surroundings. The bat algorithm developed by Yang depends on the following three assumptions:

1. The bats use echolocation to calculate the distance and differentiate between prey and objects in the background.
2. Each bat is flying randomly with velocity at position x and with a frequency f . The frequency may have varying wavelength λ and loudness A_0 to search for its prey.
3. That Bat loudness changes from a large positive value A_0 to a minimum constant value A_{min} .

Algorithm 1. Parallel Distributed Bat Algorithm: Master – Slave (PDBA-MS)

```

Initialize send_time to N
For (iterate = 1 to MaxNoIter)
IF ( Slave Node ) then
  For ( k = 1 to PopulationSize) Note: PopulationSize is N / total_slaves
    1. Perform the sequential bat's algorithm
  End
  If ( NoOf_iterations equals send_time ) then
    a. Each slave_k sends its best candidate solution to the Master.
    b. Receive the best from Master node.
  Else if ( Master Node ) then
    For ( j = 1 to Total_Slaves )
      1. Receive best_j from each slave.
      2. Compare and choose best among them.
      3. Send to each slave the best.
    End
  ENDIF

```

The algorithm is based on the hunting behaviour of bats. Bats use echolocation to detect its prey, obstacle avoidance and resting location pinpointing. Bats emit ultrasonic audio frequencies and interpret their echoes to determine the distance and direction of targets. The Bat algorithm has the following main features: automatic zooming via loudness and pulse emission rates, parameter control, frequency tuning. It has the following advantages: ability to efficiently solve a wide range of nonlinear optimization problems with optimal solutions. The Bat algorithm has proven to provide solutions in a variety of applications, namely, Engineering design, Protein Structure prediction, Classification of genes, PID (process identification) controllers, and Neural Networks. In this research, we use the Parallel Distributed Bat Algorithm (PDBA) as summarized in algorithm 1 in obtaining the optimum weights of a Deep Neural Network [10]. The parallel bat algorithm is based on the inherent behaviour of each bat echolocation. Every bat is flying independently with its own frequency, velocity and location. Therefore, a straight forward parallel method would have to have a number of processors equal to the number of flying bats. Each processor executes in a parallel and distributed fashion. However, when there are a limited number of processors, then a group of X bats or solutions will be assigned to each processor. In a parallel distributed algorithm, each worker node works on a split population (i.e. solutions) depending on the size of the cluster. The parallel-distributed bat algorithm using the Master-Worker model is summarized as follows. The process is terminated when the desired accuracy is achieved or the maximum number of iterations have been reached.

In PDBA, the master may be used to perform primarily the exploration part and the slaves primarily perform the exploitation part more than exploration of the search space. Other scenarios are possible. The main objective in the PD method is to reduce the communication time as much as possible compared to the computation time.

3. Artificial Neural Networks

Artificial Neural networks (ANN) are inspired by the biology of the human brain, which consists of neurons connected to other neurons via axons. It has been shown that the human brain consists of

10^{11} neurons. An example of how neurons are connected via axons and dendrites are shown in the Fig. 2, below. The dendrites act as inputs receiving signals from the sensors as electrical impulses and are processed by neurons.

In an ANN there are several layers of nodes and each node has an activation function which is computed and its output is passed to next layer of neurons connected to it. Each link connecting the nodes have a weight value associated with it.

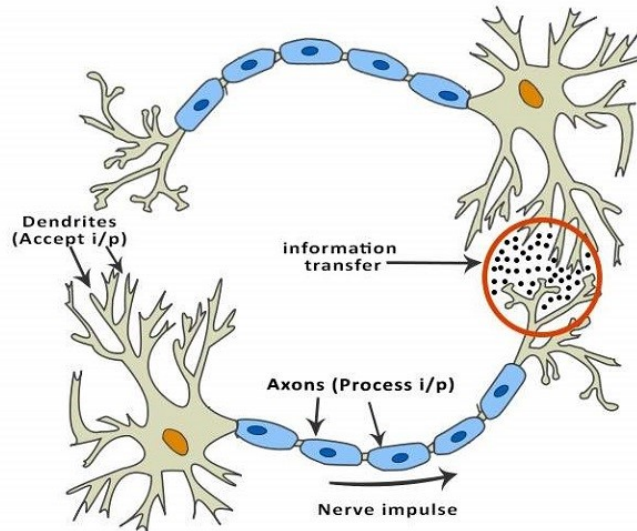


Figure 2. Biology of human brain cells² [26].

A Neural Network (NN) consists of two topologies, feedforward or feedback. In a feedforward NN as the name implies, the flow is one way without any feedback loop. These networks are beneficial in pattern generation, pattern classification and pattern recognition problems.

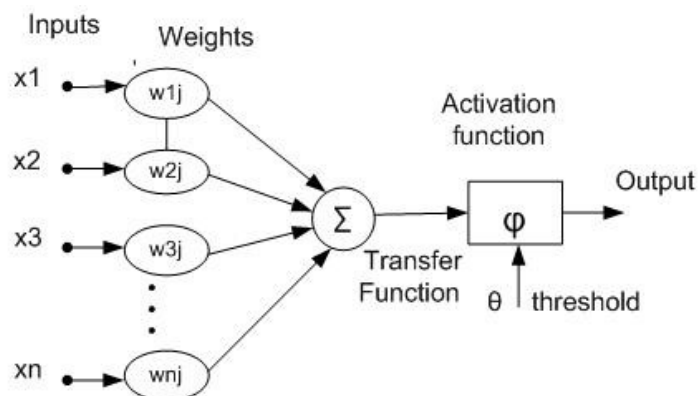


Figure 3. Neuron with Activation Function.

Neurons are connected in each layer and weights assigned to the links with an arrow indicating the movement of information, as shown in Fig. 3, above. A function at each neuron is then computed with the inputs and weights multiplied and summed. The network is trained with data and the desired outputs. In the training phase, the weights are changed incrementally such that the desired output is achieved. There are alternative methods to make the deep learning neural networks learn, namely, reinforcement learning, supervised learning and unsupervised learning. In reinforcement learning, the Neural network observes the environment, makes a decision, and the weights are adjusted if the observation is not correct. Supervised learning consists of an instructor type guiding the NN with the correct answers and adjustments to the weights are done by observing the loss function result. In unsupervised learning, no instructor is used to guide the NN with an example data set.

² https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.htm

In the feedback topology the neural network usually uses a back-propagation method with the feedforward part. In the back-propagation method, the weights are adjusted by calculating the derivatives of a loss function. The main problem with this method is convergence to a local optimum. The algorithm for deep learning neural networks with back-propagation method is summarized as follows:

Algorithm 2. For Deep Learning Neural Networks: [24]

1. Initialize: Assign random numbers to the weight matrix.
2. Propagate forward using weights and activation functions.
3. Calculate using the error function sum of squares of absolute error
4. Using differentiation of the loss error function or use optimization technique to find optimum weights.
5. Perform back-propagation of errors to the hidden layers.
6. Adjust the weights.
7. Repeat steps 2 to 6 until desired convergence is achieved.

The main objective is to design a NN for prediction of spoken words in control systems. Let F_{jk}^L be the LPC feature inputs, where L represents the NN layer number, j denotes data and k denotes input. Using the general theory of Neural Networks with back propagation as in [24], the following is presented for completeness. For the Deep Learning NN, the number of hidden layers is more than two:

$$F^1 = \begin{bmatrix} f_{11}^1 & f_{12}^1 \\ \vdots & \vdots \\ f_{N1}^1 & f_{N2}^1 \end{bmatrix} \tag{1}$$

The weight matrix at layer 1 is represented by W^1 , where W_{qr} denotes the link weight from node q to node r .

$$W^1 = \begin{bmatrix} W_{11} & \cdots & W_{14} \\ \vdots & \ddots & \vdots \\ W_{21} & \cdots & W_{24} \end{bmatrix} \tag{2}$$

Let H be the resultant matrix obtained by multiplying the input features F by the weights W matrix as shown,

$$H_1^1 = F W$$

$$H_1^1 = [F_{i1}W_{11} + F_{i2}W_{21} \quad F_{i1}W_{12} + F_{i2}W_{22} \quad F_{i1}W_{13} + F_{i2}W_{23} \quad F_{i1}W_{14} + F_{i2}W_{24}] \tag{3}$$

Let $A(v)$ be the activation function matrix, where the sigmoid function is utilized at each node, $s(v) = 1/(1 + e^{-v})$ [9]

$$A^1(v) = \begin{bmatrix} s_1(v1) & s_1(v2) & \cdots & s_1(v4) \\ \vdots & \vdots & \ddots & \vdots \\ s_N(v1) & s_N(v2) & \cdots & s_N(v4) \end{bmatrix} \tag{4}$$

Let G^2 be the output obtained by function matrix $S(v)$ multiplied by the weight matrix W^2 at layer 2, as shown,

$$G^2 = S^2(v) W^2 = \begin{bmatrix} G_{11}^2 & G_{12}^2 & \cdots & G_{14}^2 \\ \vdots & \vdots & \ddots & \vdots \\ G_{N1}^2 & G_{N2}^2 & \cdots & G_{N4}^2 \end{bmatrix} \tag{5}$$

where W^2 is the weight matrix at layer 2.

The performance of the neural network may use the squared error function:

$$Error = -\sum_{z=1}^4 0.5 x (target_z - actual_z)^2 \tag{6}$$

or the cross entropy:

$$CE = CE(target, output) = -\sum_{z=1}^4 t_{iz} \log(o_{iz}) \tag{7}$$

where t represents the target and o represents the actual output of the trained neural network. The optimal weights in the backpropagation method are computed by the gradient descent of the total error, $Error$, as given below,

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial z_k} x \frac{\partial z_k}{\partial f} x \frac{\partial f}{\partial w_i} \tag{8}$$

The cross entropy, CE , is obtained by calculating the partial derivative of the CE with respect to the weights and is given by,

$$\frac{\partial CE}{\partial w_i} = \frac{\partial CE}{\partial z_k} x \frac{\partial z_k}{\partial f} x \frac{\partial f}{\partial w_i} \tag{9}$$

After taking partial derivatives and evaluating, the old weights are computed as follows,

$$W_{new} = W_{old} - \text{delta} * \nabla_w CE \quad (10)$$

Where delta represents a small step value. The new value of weights does not guarantee it is better than the old weight, it might be necessary to re-adjust weights according to the error convergence criteria.

Neural networks are inherently parallel as the connections of elements are so in each layer. A neural network is trained for a particular input to map to a specific target output. The output of a NN often have binary values, however when implementing a speech recognition system, the problem is that there is overlapping of classes. In such a case it is better to use a probabilistic neural network in which the output values are numbers between 0 and 1, and that the sum of outputs equals one.

There are three approaches to updating the weights of a NN: online mode, batch mode and stochastic mode. In the online mode, after each training sample the weights are updated. In the batch mode, the weights are updated after all samples in the training set are run. In the stochastic mode, the updates of weights are done in mini-batches.

In practice, the backpropagation algorithm is used to update the weights of a Deep Learning NN as it performs well in challenging problems even though it is slow in convergence and tends to a local minimum. The training of a neural network is an art of balancing between the learning (training dataset) and generalization by checking its performance on new unseen inputs to a NN. In practice, backpropagation is used as a popular method due to its simplicity and computational efficiency. When implementing the backpropagation method, it is important to look at the following parameters; the three different modes of updating of the weights, randomizing the input samples, initialization of weights, choice of activation functions (e.g. Sigmoid or ReLu or others), choice of target representations (e.g. 0, 1 or -1, 1), and choice of the learning rate.

4. Application to Speech Recognition

In this section, an application to speech recognition is presented. Speech is one of the most efficient ways to communicate and convey information by humans. Currently, speech recognition systems are used to replace input devices such as mouse and keyboards. Speech recognition is also a very desirable way to communicate with machines such as robots.

There are many methods used for speech recognition [13]. The voice signals are sampled at 8000 samples/s directly from the microphone and LPC is used for extracting 13 features from a human voice signal. Many samples of a particular word for neural network to recognize are collected and stored in matrices of arrays. These samples are then used to train a Deep Learning Neural Network to recognize the uttered word. After training is done and optimum weights of the Neural Network determined, the NN network is then tested to check for satisfactory results. The Bat algorithm with the Backpropagation method were used to find the optimum weights of the Neural Networks. If the results are not satisfactory then Deep Learning NN is re-trained. The architecture of the proposed parallel neural network with the Bat optimization algorithm is shown in Fig. 4.

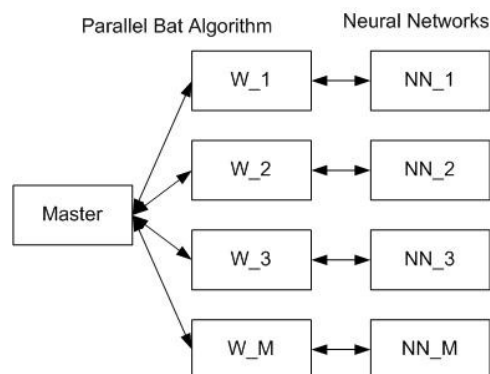


Figure 4. Parallel Neural networks with Parallel Distribute Bat optimization algorithm for weight determination.

Algorithm 3. Speech Recognition Method

- Step 1.** Speak clearly the words to be trained by neural network.
 - a. Around 13 features of the spoken word are extracted.
 - b. Due to memory limitation, only 10 words were used in the experiment.
- Step 2.** Run the neural network with weights and calculate the outputs and error.
 - a. Neural network with backpropagation and evolutionary Bat Algorithm were implemented on Arduino microcontrollers
- Step 3.** Test the neural network on microcontrollers. Speak into the microphone the words to be tested.

The following 14 words were tested using algorithm 3: clear, open, close, door, window, left, right, up, down, stop, go, on, off, and motor. The pre-processing of speech was performed using MATLAB® and some sample figures are presented below as Fig. 5-7, showing the LPC values of 13 features. Each figure in Fig. 5-, represents six samples of each word spoken. Fig. 8 shows the error dropping per iteration for *N* generations. Table 2 depicts the words used in the experiment and the percentage of time the neural net recognized the words correctly. Although the majority of the words are in the 90% range, higher accuracy is required for critical applications.

Table 2. Percentage of time the machine recognized the words correctly.

#	Trained Words	Correct (%)	#	Trained Words	Correct (%)
1	Clear	93	8	up	89
2	Open	91	9	down	9
3	Close	94	10	stop	93
4	Door	90	11	go	92
5	Window	88	12	on	89
6	left	91	13	off	96
7	right	93	14	motor	91

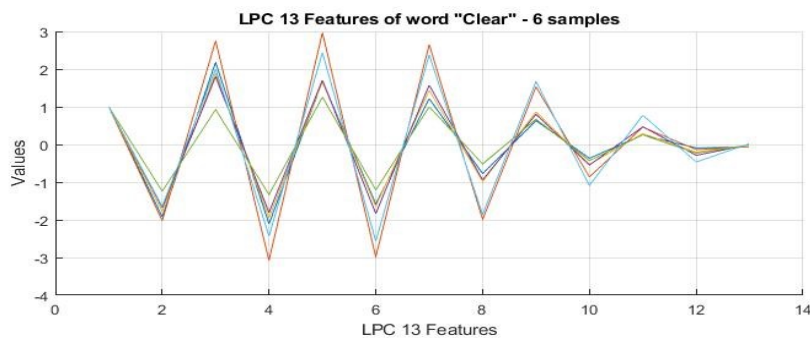


Figure 5. LPC 13 features of the word “Clear”.

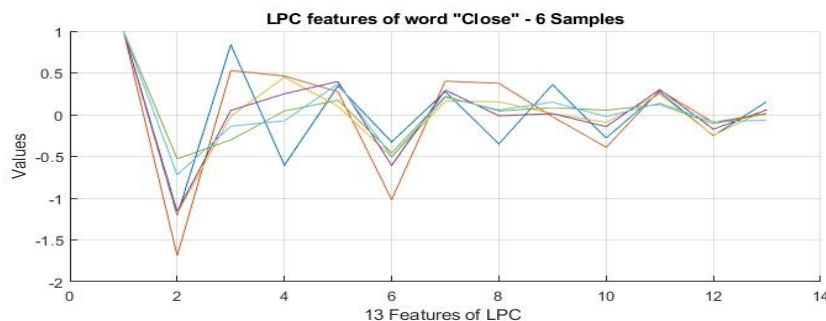


Figure 6. LPC 13 features of the word “Close”.

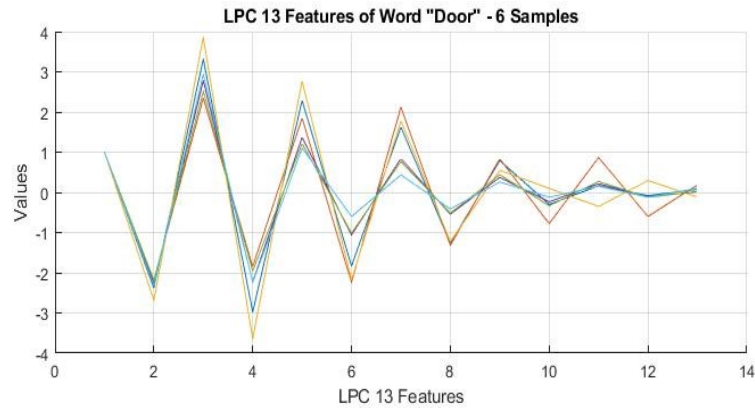


Figure 7. LPC features of the word "Door".

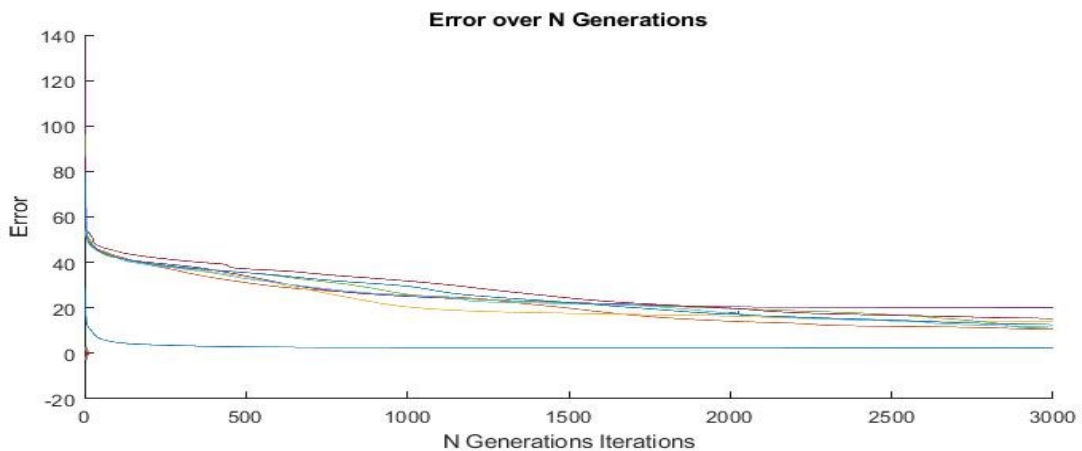


Figure 8. Error per iterations for N generations.

5. Conclusion

A novel hybrid method consisting of combining Parallel Distributed Bat Algorithm with Backpropagation method was presented. It is well known that usage of the backpropagation method often produces results converging to a local minimum, however, the hybrid method proposed avoids such a situation. The experimental part consisted of implementing parallel neural networks using several Arduino microcontrollers for faster convergence and testing an application. The weights of the neural networks were computed via a combination of the Backpropagation method and the parallel distributed Bat algorithms. The purpose of the Bat algorithm is to help avoid local convergence. Application to speech recognition was tested and the performance of the neural network was observed. As Arduino microcontrollers have limited hardware resources, the MATLAB® software was used to perform the pre-processing of voice commands and only the LPC parameters were passed to the Neural nets implemented on the Arduino® microcontrollers. Three Arduino® microcontrollers each having a neural network algorithm with selected words were tested. The performance was measured by observing the accuracy of speech commands interpreted by the neural nets.

It was observed that high level of classification accuracy by the neural network of spoken words were achieved in the 90% range and therefore making it attractive for voice-controlled applications. In speech recognition, it is desirable to have an accuracy near 99% for more critical voice-controlled applications and this is the subject for future research.

Acknowledgment

This research was supported by the Takamul Research Grant #25/40 from the Deanship of Scientific Research from the Islamic University of Madinah, Kingdom of Saudi Arabia. The authors

are also grateful to the Faculty of Computer Science and Information Systems at the Islamic University of Madinah for their support in this research. The authors are also very grateful to the comments of all the reviewers which included extensive proofreading which have all helped improve the paper substantially.

References

- [1] Xin-She, Y. (2011). Bat algorithm for multi-objective optimization. *Internal Journal of Bio-Inspired Computation*, vol. 3, no. 5, pp. 267-274.
- [2] Yang, X.S. (2010). A New Metaheuristic Bat-Inspired Algorithm. In: González J.R., Pelta D.A., Cruz C., Terrazas G., Krasnogor N. (eds). *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. *Studies in Computational Intelligence*, vol 284. Springer, Berlin, Heidelberg pp. 65-74.
- [3] Hagan, M.T., Demuth, H.B., Beale, M.H., DeJesus, O. (2014). *Neural Network Design*, 2nd Edition, eBook available from: <https://hagan.okstate.edu/NNDesign.pdf> [accessed 17 May 2020].
- [4] Zipser D., Andersen, R.A. (1988). A Back Propagation Programmed Network that Simulates Response Properties of a Subset of Posterior Parietal Neurons. *Nature*, vol. 331, pp. 679-684.
- [5] Chan, L.-W., Fallside, F. (1987). An Adaptive Training Algorithm for Back-Propagation Networks. *Comp. Speech and Language*, vol. 2, Issues 3-4, Sept. – Dec. 1987, pp. 205-218.
- [6] Noor, F., Ibrahim, A., AlKhattab, M.M. (2020). Performance of Parallel Distributed Bat Algorithm using MPI on PC Clusters. *Annals of Emerging Technologies in Computing*, vol. 4, no. 1, Jan 1 2020, pp. 19-27.
- [7] Noor, F. *Pattern recognition using Neural Networks*. Project. Islamic University of Madinah, 2018.
- [8] Nagy, G. (1983). *Candide's Practical Principles of Experimental Pattern Recognition*. *IEEE Trans. on Pattern Anal. and Machine Intel.*, vol. PAMI-5, no. 2, pp. 199-200, March 1983.
- [9] Tenorio, M.F., Lee, W.T. (1989). Self Organizing Neural Networks for the Identification Problem. In *Advances in Neural Information Processing Systems*, CA, San Mateo: Morgan Kaufman, vol. 1, pp. 57-64.
- [10] Lippmann, R.P. (1989). Pattern Classification using neural networks. *IEEE Communications Magazine*, vol 27, no. 11, Nov pp. 47-50.
- [11] Sang Y., Shi, L., and Liu, Y. (2018). Micro Hand Gesture Recognition System Using Ultrasonic Active Sensing. *IEEE Access*, vol. 6, pp. 49339 – 49347.
- [12] Mora, J. (2017). Chirp Microsystems Sticks Sonar in Chips That Detect the World Around Them. *Electronic Design*, Dec 19. Available: <https://www.electronicdesign.com/embedded-revolution/chirp-microsystems-sticks-sonar-chips-detect-world-around-them>. [Accessed: 17 May 2020].
- [13] Hruschka, E.R., and Ebecken, N. (2003). A genetic algorithm for cluster analysis. *Intelligent Data Analysis*, vol. 7, no. 1, Feb 2003, pp. 15-25. Available: https://www.researchgate.net/publication/220571471_A_genetic_algorithm_for_cluster_analysis. [Accessed 17 May 2020].
- [14] Huang, D., and Lin, J. [2010]. Scaling Populations of a Genetic Algorithm for Job Shop Scheduling Problems Using MapReduce. 2010 IEEE Second Int. Conf. on Cloud Computing Technology and Science, Indianapolis, IN, 2010, pp. 780-785, doi: 10.1109/CloudCom.2010.18.
- [15] Misbahuddin, S., and Noor, F. (2007). *Hands-on Workshop on Parallel Processing*. University of Hail, Kingdom of Saudi Arabia..
- [16] Ma, X., Zhang, J., Du, B., Ding, C., and Sun, L. (2019). Parallel Architecture of Convolutional Bi-Directional LSTM Neural Networks for Network-Wide Metro Ridership Prediction. *IEEE Trans. on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2278-2288, June 2019, doi: 10.1109/TITS.2018.2867042.
- [17] Wang, F., Yu, P.L.H., and Cheung, D.W. (2014). Combining technical trading rules using parallel particle swarm optimization based on Hadoop. 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, 2014, pp. 3987-3994, doi: 10.1109/IJCNN.2014.6889599.
- [18] Zhao, W., Ma, H., He, Q. (2009). *Parallel K-Means Clustering Based on MapReduce*. *CloudCom 2009. Lecture Notes in Computer Science*, vol 5931. Springer, Berlin, Heidelberg, pp. 674-679.
- [19] Wilkinson, B., and Allen, M. (2005). *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*. 2nd Edition. Pearson.
- [20] Chandra R., Dagum, L., Kohr, D., Maydan, D., McDonald, J., and Menon, R. (2001). *Parallel Programming in OpenMP*, San Francisco, CA: Morgan Kaufmann Publishers. Available: https://apps2.mdp.ac.id/perpustakaan/ebook/Karya%20Umum/Parallel_Programming_in_OpenMP.pdf [accessed: 17 May 2020].
- [21] Wellekens, C.J. (1998). *Introduction to Speech Recognition Using Neural Networks*. *ESANN'1998 Proceedings - European Symposium on Artificial Neural Networks*, Bruges (Belgium), 22-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 227-236. Available: <https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es1998-456.pdf> [accessed: 17 May 2020].
- [22] Jalali, B., Chen, C.L., and Mahjoubfar, A. (2017). Cancer Diagnostics with Deep Learning and Photonic Time Stretch. *MathWorks Technical Articles and Newsletters*, published online 2017 – 93090v00. Available:

<https://www.mathworks.com/company/newsletters/articles/cancer-diagnostics-with-deep-learning-and-photonic-time-stretch.html> [accessed: 17 May, 2020].

- [23] Nawi N.M., Rehman M.Z., Khan A. (2014). A New Bat Based Back-Propagation (BAT-BP) Algorithm. In: Swiątek J., Grzech A., Swiątek P., Tomczak J. (eds)., *Advances in Systems Science. Advances in Intelligent Systems and Computing*, vol. 240. Springer, Cham. Available: https://doi.org/10.1007/978-3-319-01857-7_38.
- [24] Noor, F. [2019]. Pattern Recognition using a Neural Network on a Microcontroller with I2C Ultrasonic Sensors. *Annals of Emerging Technologies in Computing (AETiC)*, pp. 9-19, vol. 3, no. 1, 1st Jan 2019.



© 2020 by the author(s). Published by *Annals of Emerging Technologies in Computing (AETiC)*, under the terms and conditions of the Creative Commons Attribution (CC BY) license which can be accessed at <http://creativecommons.org/licenses/by/4.0>.