Fuzzy Systems and Data Mining VIII
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA220372

Fractional Gradient Descent Learning of Backpropagation Artificial Neural Networks with Conformable Fractional Calculus

Mohammad Rushdi Saleh and Basem Ajarmah[1]
Information Technology, Al-Istiqlal University, Jericho, PALESTINE

Abstract. Conformable fractional calculus will be a promising area of research for information processing as natural language and material modelling due to its ease of implementation. In this paper, we propose a fractional gradient descent method for the backpropagation training of neural networks. In particular, the conformable fractional calculus is employed to evaluate the fractional differential gradient function instead of the classical differential gradient function. The results obtained on a large dataset with this approach provide a new optimized, faster and simpler implemented algorithm than the conventional one.

Keywords. Neural networks, material modelling, gradient descent, backpropagation, fractional differential gradient, conformable fractional calculus, natural language, optimization.

1. Introduction

Artificial Neural Networks (ANN) are computational model based on human brain function and nervous system. This Model was explored by Warren McCullouch and Walter Pitts in (1943) [1], by creating a computational model based on algorithm called threshold logic. Werbos (1975) [2] was renewed the interest in neural networks and learning by introducing backpropagation algorithm which enable practical training of multi-layer networks.

ANN model has proposed in many disciplines for classification, clustering, pattern recognition and prediction. ANN is a powerful data processing tool with high classification accuracy and a strong ability to process data in parallel. Though, the great potential of ANNs is the high-speed processing provided in a massive parallel implementation. ANN applications can be evaluated with respect to data analysis factors such as accuracy, processing speed, latency, fault tolerance, volume, scalability and convergence and performance. [2,3]

The main idea stand behind how neural network learned is the propagating information through one or multiple layers of neurons. Hence, a mathematical function used by each neuron to process information. To reach the expected outcome a set of

[1] Corresponding Author, Basem Ajarmah, Information Technology, Al-Istiqlal University, Jericho, PALESTINE; E-mail: basem@pass.ps.

weights added to the input information and iteratively adjusted through a process called backpropagation. The goal of backpropagation is to adjust the weights and biases to the neural network by calculating the cost, where the cost will minimized in the next iteration. This process repeated to find the lowest value of cost function.

Backpropagation algorithm works by calculating the gradient of the loss function, which lead us to the value that minimize the loss function. However, by using gradient descent we can iteratively move to the minimum value toward the direction given by the gradient.

One of the challenges of the artificial neural networks modelling which received attraction of researchers and require more investigations is the continuous gradient puzzle and quantization of variable problems and noise.

The gradient mainly depends on the randomly weights which assigned to the features, the point here is to optimize weight to minimum error to achieve minimum value of the loss function. Obviously, the error will vary according to the weight, so, it used the derivative of the error with respect to the weight. Where, this derivative called gradient.

Boroomand [4], Kaslik [5], Pu [6], Wang [7], and Bao [8] have successfully used the fractional derivative on artificial neural networks. And all this was done because the fractional derivative has a dynamism in the application and the ability to reduce and improve the weight to a minimum error to achieve the minimum value of the loss function. But all of this research was done using Caputo's definition. Our work is to use another definition of a fractional derivative, which is conformable Fractional Derivative CFD.

In this paper, we will apply CFD in minimization work. Minimization can be done in several aspects because the fractional derivative can be easily applied using fractional Newton's method, fractional loss function, and derivative of the total error with respect to the weight. The performance of the proposed models respectively was evaluated on the MINST hand writing dataset with three-layer network, Synthetic spiral data set with three-layered network, and MINST hand writing dataset with eight-layer network.

The structure of the paper is as follows: in Section 2, Algorithms Description using CFD conformable fractional calculus are introduced. In Section 3, experimental results are presented to illustrate the proposed fractional-order methods in layered BP neural networks. Finally, the paper is concluded in Section 4.

2. Algorithms Description using CFD

Fractional calculus is the most developed area of optimization and minimization problems [9]. Because of the difficulty and complexities of applying this type of derivation or integration; many definitions of fractional derivative or integration appear, such as Riemann-Liouville, Caputo's, CFD, and etc. There is a trend to explore and create new definitions and models for fractional differential operators. Also there is a desire to impose strict criteria and definitions of what we call “Fractional derivative” or “Fractional integral” [10-12].

There are a large number of problems that require finding within it the Gradient Descent (steepest descent). Gradient Descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. The idea is to take repeated steps in the opposite direction of the approximate gradient of the function at the current point, in order to find the direction of steepest descent. on the other hand, stepping in the direction of the gradient will lead to a local maximum of that function.

Fractional operator can be inserted into any part or section of the problem, so we decided to do it in several ways, which are Newton's method, loss function, and derivative of the total error with respect to the weight.

Khalil et al. [13], proposed a new definition of fractional derivative Eq. 1. It's based on new relation between fractional calculus and calculus based on basic limit definition.

$$f^{\gamma}(t) = \lim_{\tau \to 0} \frac{f\left(t+\tau t^{1-\gamma}\right)-f(t)}{\tau}, \forall t > 0, 0 < \gamma < 1 \quad (1)$$

Where γ – fractional order, f^{γ} the fractional derivative notation as D^{γ} , if γ = 1 then f' is the classical derivative, and when γ = 0.5 then it represent fractional derivative with 0.5 as fractional power [14]. that implies:

$$f^{\gamma}(t) = D^{\gamma} f(t) = t^{1-\gamma} \frac{d}{dt}\left(f(t)\right) = t^{1-\gamma} f'(t) \quad (2)$$

2.1. Fractional Newton's Method

Newton's method attempts to solve optimization on differentiable function mainly loss error

$$Loss = E_j = 0.5\left(\sum_{i=1}^{n^{(L)}} (a_{ji}^{(L)} - o_{ji})^2\right) \quad (3)$$

Where $a_{ji}^{(L)}$ denotes the i-th element of $A_j^{(L)}$ is the total neuron in the layer, using Taylor's expansion approximation performs in the iteration take the form [15].

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)} \quad (4)$$

The fractional form of Eq. 4 using CFD is

$$x_{i+1} = x_i - \frac{x_i^{1-\gamma} f'(x_i)}{f''(x_i)} \quad (5)$$

If we define learning rate as Eq. 6

$$Learning\ rate = \frac{x_i^{1-\gamma}}{f''(x_i)} \quad (6)$$

Also $f''(x_i) = 1$, for E_j loss error for output layer, learning rate = $x_i^{1-\gamma}$. For example if fractional order 0.5 the learning rate $\eta = \sqrt{x_i}$, Eq. 4 became gradient descent for output layer

$$x_{i+1} = x_i - \sqrt{x_i}\, f'(x) \quad (7)$$

So the updated weights for γ = 0.5

$$(w_{ji}^l)^{t+1} = (w_{ji}^l)^t - \sqrt{w_{ji}^l}\ \frac{\partial E_j}{\partial w_{ji}^l} \quad (8)$$

In this algorithm we just only find formula for learning rate η. As The fractional derivatives ensure noise resilience [16].

2.2. Fractional loss function on delta

In order to minimize the total error of the fractional order BP neural network; we need to estimate the fractional CFD gradient descent for output layer. So we need to find delta δ_i^l which defined as

$$\delta_i^l = \frac{\partial E_j}{\partial z_i^l} \tag{9}$$

Where z_i^l inputs of z-th layer. E_j is defined in Eq. 3, the fractional form of CFD delta δ_i^l will be

$$\delta_i^l = \frac{\partial^\gamma E_j}{\partial (z_i^l)^\gamma} = (z_i^l)^{1-\gamma} \frac{\partial E_j}{\partial z_i^l} \tag{10}$$

According to Eq. 3, and using chain rule for output layer delta we can now write that[13]

$$\delta_i^L = \frac{\partial^\gamma E_j}{\partial (z_i^L)^\gamma} = (z_i^L)^{1-\gamma} \sum_{j=1}^{L} (-a_{ji}^{(L)} + o_{ji}) f_L^{'}(z_i^L) \tag{11}$$

Where f' denotes the corresponding any type of activation function derivative for the L-th layer, and denotes the i-th element of O_j that are the input and the corresponding ideal output of the j-th sample [8]. If the activation function of output layer is sigmoid then output delta layer will be

$$\delta_i^L = (o_{ji})^{1-\gamma} \sum_{j=1}^{L} (a_{ji}^{(L)} - o_{ji}) (o_{ji}(1 - o_{ji})) \tag{12}$$

Then the relationship between δ_i^l and δ_i^{l+1}can be given by

$$\delta_i^l = \frac{\partial E_j}{\partial z_i^l} = \sum_{j=1}^{n^{l+1}} \delta_i^{l+1} w_{ji}^l f_L^{'}(z_i^l) \tag{13}$$

We just generate fractional loss function in the output layer

2.3. CFD total error with respect to the weight

The fractional updating formula of total error with respect to weight is [8,17]:

$$(w_{ji}^l)^{t+1} = (w_{ji}^l)^t - \mu \frac{\partial^\gamma E_j}{\partial (w_{ji}^l)^\gamma} \tag{14}$$

Using CFD Eq. 14 became based on Eq. 2

$$(w_{ji}^l)^{t+1} = (w_{ji}^l)^t - \mu (w_{ji}^l)^{1-\gamma} \frac{\partial E_j}{\partial w_{ji}^l} \tag{15}$$

And

$$\frac{\partial E_j}{\partial w_{ji}^l} = \delta_j^{l+1} a_i^l \tag{16}$$

If you need to override overfitting problem that raises when trying to test your structure model with testing dataset, then you must introduce a regularization L term to the error loss

$$E_L = E + \frac{\lambda}{2} \|W\|^2 \tag{17}$$

Where $\|W\|^2$ the sum of squares of all weights and λ>0 regularization parameter. That update Eq. 15

$$(w_{ji}^l)^{t+1} = (w_{ji}^l)^t - \mu \left(\left(w_{ji}^l\right)^{1-\gamma} \frac{\partial E_j}{\partial w_{ji}^l} - \lambda \left(w_{ji}^l\right)^{2-\gamma} \right) \tag{18}$$

you will notice that solving problem with CFD method is much simpler and less computational memory and cost than other fractional definition as Caputo or Riemann-Liouville [18,19].

3. Experiments

In this section, the following three simulations were carried out to evaluate the performance of the presented algorithms.

3.1. Fractional Newton's Method simulation

The simulation has been performed on the MNIST handwritten digital dataset. Each digit in the dataset is a 28×28-pixel image. Each image is associated with a label from 0 to 9. We represented dataset consist of 42,000 images (training sample). Each image represented as an array of 784 elements. Each element has a value from 0 (completely full black) to 255 (completely full white); then add an element to represents the number written in this image. So we have matrix with 42000 row representing 42000 images. Each row consists of 785 columns. All element divided by 255 Except for the last column divided on 10.

We divided this dataset into two sets the training set 0.8 of the hole set (33600 row), and the testing set of 0.2 of set (8400 row). In order to identify the handwritten digits in MNIST dataset, a neural network with 3 layers. the input layer consists of 784 value. The hidden layer has 10 neurons and 10 biases and output layer of one neuron and one bias. We use sigmoid activation function in all layers. we use Eq. 8 that have fractional learning rate to update the weights (784 W for inputs plus 10 bias for hider layer and 10 W plus 1 bias for output) using fractional order of 0.5; in this algorithm the learning rate changes as square root of each neuron as the fractional newton method suggest in section 2.1.

The performance of the proposed fractional-order BP neural networks with fractional learning rate and the performance comparison with integer-order BP neural networks, in terms of training and testing accuracy with the iterations are shown in Figure 2 and loss in Figure 1.

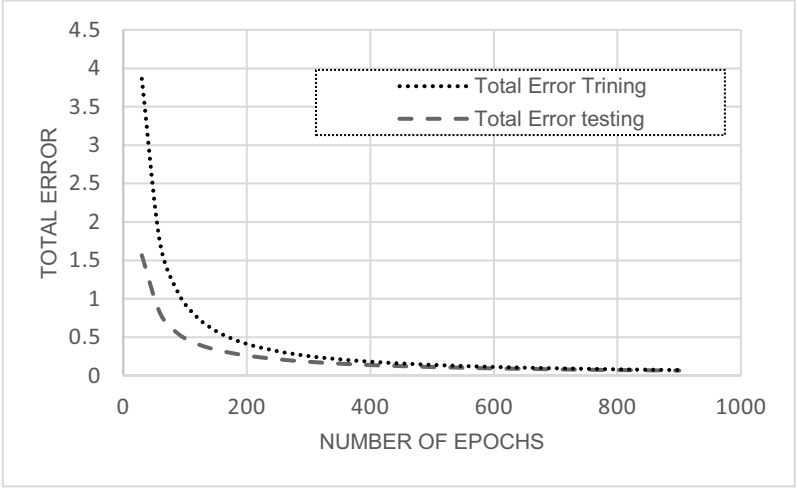

Figure 1. Changes of total error with fractional learning rate

Figure 1 represent the changes of total error with fractional learning rate in Eq. 8 that shows the stability and convergence of the proposed fractional-order BP neural networks

learning rate formula in Eq. 8. And this stability convergence happened in Figure 2 that represent Changes of Accuracy with fractional learning rate.

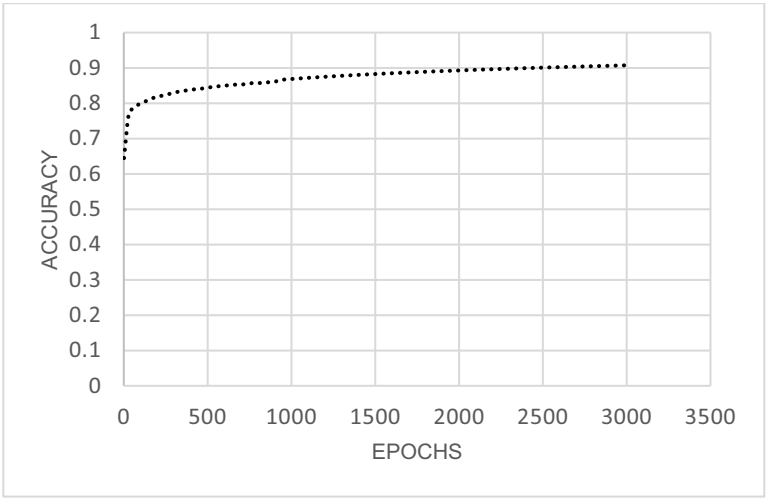

Figure 2. Changes of Accuracy with fractional learning rate Eq. 8 equal to $\sqrt{w_{ji}^{l}}$

3.2. Fractional loss function simulation

The neural network architecture topological structure is represented in Figure 5. Synthetic spiral data set with five-layered network

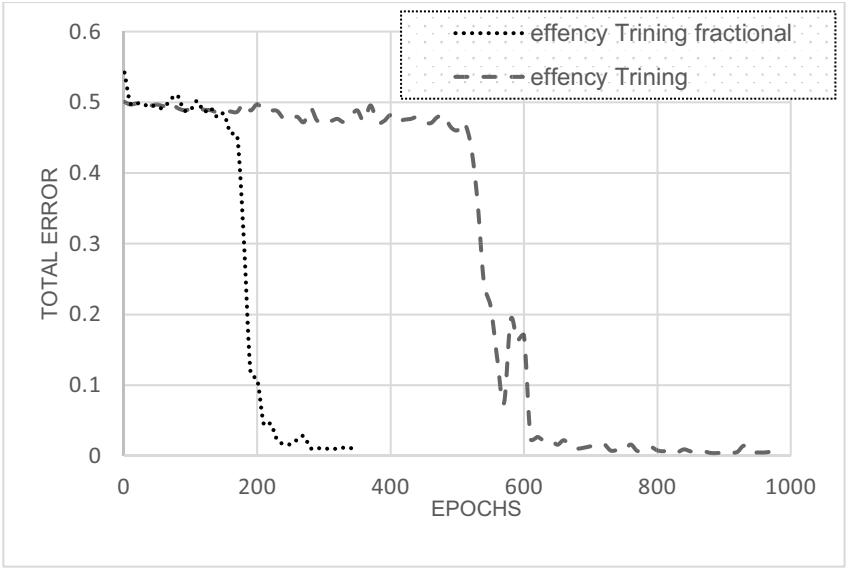

Figure 3 Changes of total error with Fractional loss function on delta

Figure 3 represent the changes of total error with fractional loss function on delta in Eq. 13 that shows the stability and convergence of the proposed fractional-order BP neural

networks fractional loss formula in Eq. 13. And this stability convergence happened in Figure 4 that represent Changes of Accuracy with fractional loss function.

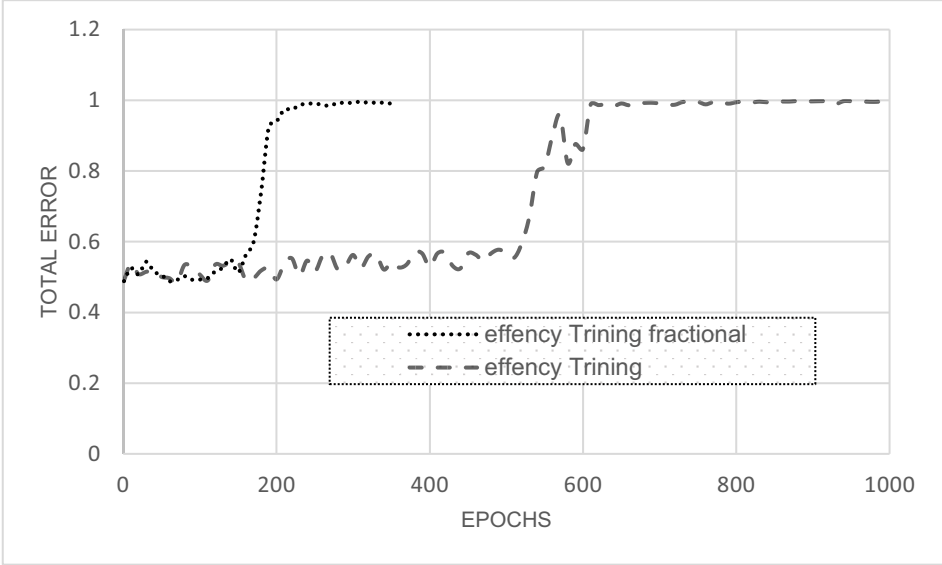

Figure 4. Changes of Accuracy with Fractional loss function on delta

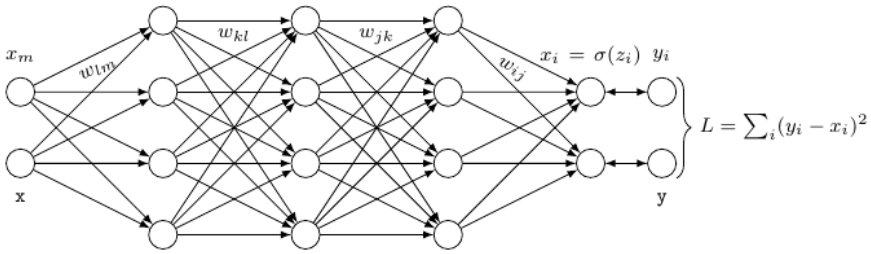

Figure 5. The topological structure of the neural networks

3.3. CFD total error with respect to the weight simulation

The third part was already tested in [8] when they used Caputo's fractional definition in neural networks using backpropagation. Where we found almost the same equations in their research with a slight difference in the factors added by the Caputo's method

4. Conclusion

In this study, fractional operator inserted into different part of the BP neural networks algorithm, we implement fractional operator using Newton's method, loss function, and derivative of the total error with respect to the weight. We proposed a new three different

modifications on three part of the backpropagation algorithm Eq. 8, Eq. 13, and Eq. 18. These proposed modification turns out that general fractional CFD method can converge to the real extreme point. The results obtained on a large dataset with this approach provide a new optimized, faster and simpler implemented algorithm than the conventional one. numerical results show that CFD very straightforward to implement. All you need just multiply the classical derivative with term $t^{1-\gamma}$. Finally, we think that the proposed fractional procedure is valuable and can be easily introduced in the Artificial neural network or deep learning. We believe that if you combine the three proposed modifications to the model at once, it may give better results; This will be our objective in future work.

References

[1] McCulloch, W. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4), pp.115-133.

[2] Werbos, P.J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences (1975).

[3] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263-1284.

[4] Boroomand, A., Menhaj, M.B. (2009). Fractional-Order Hopfield Neural Networks. In: Köppen, M., Kasabov, N., Coghill, G. (eds) Advances in Neuro-Information Processing. ICONIP 2008. Lecture Notes in Computer Science, vol 5506. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-02490-0_108

[5] Kaslik, E. and Sivasundaram, S, "Dynamics of fractional-order neural networks," in Proceedings of the 2011 International Joint Conference on Neural Network, IJCNN 2011, pp. 611–618, usa, August 2011.

[6] Pu, Y.-F, Yi, Z, and Zhou, J.-L., "Fractional Hopfeld neural networks: fractional dynamic associative recurrent neural networks," IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2319–2333, 2017.

[7] Wang , J., Wen, Y., Gou, Ye, Y., Z., and Chen, H., "Fractional-order gradient descent learning of BP neural networks with Caputo derivative," Neural Networks, vol. 89, pp. 19–30, 2017.

[8] Bao C., Pu Y., Zhang Y. Fractional-order deep backpropagation neural network. Comput Intell Neurosci . 2018, article 7361628

[9] de Oliveira, E. C, Machado, J. T. A Review of Definitions for Fractional Derivatives and Integral. Mathematical Problems in Engineering, Article ID 238459, 6 pages, vol. (2014).

[10] Yong, Zhou Basic Theory of Fractional Differential Equations (Xiangtan University, China), JinRong Wang (Guizhou University, China) and Lu Zhang (Xiangtan University, China) 2ed edition December 2016

[11] Baleanu, D., Diethelm , K., Scalas, E., Trujillo J. J. Fractional Calculus: Models and Numerical Methods, vol. 3 of Series on Complexity, Nonlinearity and Chaos, World Scientific, Singapore, (2012

[12] Ünal, E, Gökdoğan, A. Solution of conformable fractional ordinary differential equations via differential transform method, Optik, vol. 128, 264-273 (2017).

[13] Khalil, Roshdi, Horani, M. Al Horani, Yousef, Abdelrahman, Sababheh, M. A new definition of fractional derivative Journal of Computational and Applied Mathematics 2014

[14] Thabet Abdeljawad, On conformable fractional calculus Journal of Computational and Applied Mathematics Elsevier 1 May 2015

[15] Nocedal, Jorge; Wright, Stephen J. Numerical Optimization. Springer-Verlag (1999).

[16] Arora, S., Mathur, T., Agarwal, S., Tiwari, K., & Gupta, P. (2022). Applications of fractional calculus in computer vision: A survey. Neurocomputing, 489, 407-428.

[17] Singh, N., Arora, S., Mathur, T., Agarwal, S., & Tiwari, K. (2021). Stock Price Prediction using Fractional Gradient-Based Long Short Term Memory. In Journal of physics: Conference series (Vol. 1969, No. 1, p. 012038). IOP Publishing.

[18] Dong, Y., Liao, W., Wu, M., Hu, W., Chen, Z., & Hou, D. (2022). Convergence analysis of Riemann‐Liouville fractional neural network. Mathematical Methods in the Applied Sciences, 45(10), 6378-6390.

[19] Wang, X., Fečkan, M., & Wang, J. (2021). Forecasting Economic Growth of the Group of Seven via Fractional-Order Gradient Descent Approach. Axioms, 10(4), 257.