Improving Deep Learning Performance Using Random Forest HTM Cortical Learning Algorithm

Mohamed AbdElhamid Abbas , PhD
Dept. of computers and communications, Faculty Of Eng.
Delta Univ. for Science and Technology
Gamasa City, Egypt
mohamed2abbas@gmail.com

***Abstract*—Deep Learning is an artificial intelligence function that imitates the mechanisms of the human mind in processing records and developing shapes to be used in selection construction.The objective of the paper is to improve the performance of the deep learning using a proposed algorithm called RFHTMC. This proposed algorithm is a merged version from Random Forest and HTM Cortical Learning Algorithm. The methodology for improving the performance of Deep Learning depends on the concept of minimizing the mean absolute percentage error which is an indication of the high performance of the forecastprocedure. In addition to the overlap duty cycle which its high percentage is an indication of the speed of the processing operation of the classifier. The outcomes depict that the proposed set of rules reduces the absolute percent errors by using half of the value. And increase the percentage of the overlap duty cycle with 15%.**

Keywords—Deep learning; Random Forest algorithm; HTM algorithm; mean absolute percentage error; duty cycle

I. INTRODUCTION

Deep learning is portion of a widerthoughtful of relatives of system learning methods based on representations of facts. Its gaining knowledge is supervised, semi-supervised or unsupervised [1]. Deep getting to know fashions are loosely associated with facts processing and communication styles in a biological fearful machine, including neural coding that attempts to outline a courting among various stimuli and related neuronal responses within the mind [2]. Deep understanding styles with deep neural networks, deep acceptance networks and regular neural networks have been carried out to fields consisting of speech reputation, natural language processing, social community filtering [3]. One of the fundamental functions of unsupervised getting to know is to offer exact representations for facts, that can be used for detection, popularity, prediction, or visualization.

Precise representations do away with beside the point variability of the input information, while preserving the facts this is useful for the remaining venture. One purpose for the modern recovery in unsupervised gaining knowledge is the ability to deliver deep function hierarchies through way of stacking unsupervised modules on pinnacle of every other [3]. The unsupervised module at one diploma in the hierarchy is fed with the illustration vectors produced with the aid of using the level below. Higher degree representations capture excessive stage dependencies between input variables, thereby enhancing the capability of the device to grasp underlying regularities within the statistics. The output of the final layer within the hierarchy can be fed to a conventional supervised classifier. From these views, increasing the performance of deep learning is an important topic for better knowledge gain and efficient data classifications. Learning algorithms can effectively increasing the performance of deep learning.One of the most important algorithm is Random Forest. Sparse illustration has proven massive ability capabilities in dealing with these issues. Random forest is an indicator term for an ensemble of selection bushes. A simplified random forest is depicted in figure1[4]. Where the wooded area chooses the type having the most votes. If the variety of cases within the schooling set is N, then pattern of N instances is taken at random, but with replacement. This model may be the training set for rising the tree. If there are M enter variables, a number of m << M is particular such that at every node, m variables are decided on at random out of the M and the high-quality split on these m is used to split the node.

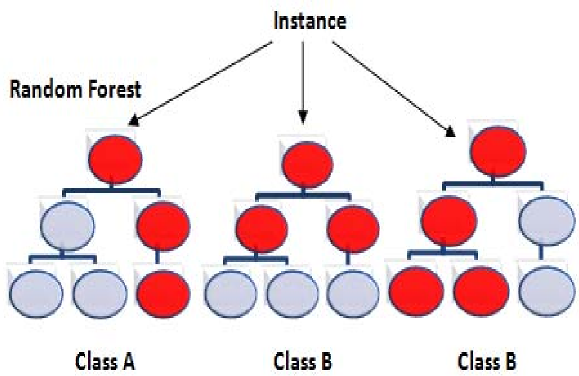

Fig.1 Simplified Random Forest [4]

Another learning algorithm that affects the performance of deep learning is Hierarchical Temporal memory (HTM) is a device studying technology that goals to capture the structural and algorithmic residences of the neocortex. The neocortex is the seat of wise idea within the mammalian mind. excessive stage imaginative and prescient, hearing, contact, motion, language, and making plans are all finished with the aid of the neocortex. Given the sort of diverse suite of cognitive features, you would possibly assume the neocortex to put into effect an equally various suite of specialized neural algorithms. HTM

978-1-5386-5083-7/18/$31.00 ©2018 IEEE

presents a theoretical framework for knowledge the neocortex and its many skills. Thus far we've got implemented a small subset of this theoretical framework. through the years, increasingly of the principle may be carried out. Figure 2 depicts the regions of HTM algorithm arranged in a four-stage chains[5].

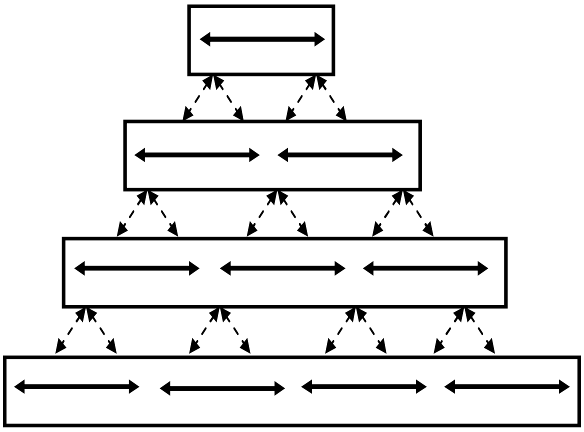

Fig.2 HTM regions arranged in a four-level Chains[5]

The main objective of this paper is to improve the performance of deep learning by proposing a new algorithm. The proposed algorithm is a combined version from random forest and HTM Cortical algorithm. The proposed algorithm is called Random Forest HTM Cortical Learning Algorithm or shortly RFHTMC. The rest of the paper is prearranged as follows: Section II initiates a literature review about deep learning preceding work. Section III is a formulation of the research problem. Section IV explains the proposed methodology including a detailed description about RFHTMC algorithm. Section V presents the simulation results of RFHTMC learning algorithm and a discussion about its impact on the performance of deep learning.

II. Literature Review

A number of researches have been concerned with describing the performance of deep learning. A unique item localization technique is introduced in [6] with the reason of boosting the localization accuracy of kingdom of-the-artwork object detection systems. D. P. Kingma in [7] introduces a set of rules for first-order gradient-based totally optimization of stochastic goal capabilities, based totally on adaptive estimates of decrease-order moments. The approach is straightforward to put in force, is computationally efficient, has little memory necessities, is invariant to diagonal re-scaling of the gradients, and is properly suitable for problems which are large in phrases of facts and parameters. Adeep assessment network to conquer the aforementioned barriers is added in [7]. This deep community includes complementary additives, a pixel degree completely convolution flow and a section-sensible spatial pooling flow.

Nian Liu and Junwei Han in [8] propose a novel quit-to-surrender deep hierarchical saliency network (DHSNet) based totally on neural networks for detecting salient-gadgets. DHSNet first makes a rough worldwide prediction via mechanically learning various worldwide dependent saliency cues, including international assessment, compactness, and their most appropriate aggregate. In [9] a comprehensive study is provided to evaluate the performance of deep gaining knowledge of above all based face representation under several situationstogether with the combination of head front angles, alteringenlightenment of different strengths, and misalignment because of erroneous facial function localization. A. Sharma in [10] analyze the getting to know of RCPN parameters and find out the presence of error paths in the computation graph of RCPN that could hinder background propagation. Naiyan Wang, Dit-Yan Yeung in [11] observe the hard trouble of tracking the trajectory of a moving object in a video with possibly very complex background. In comparison to most present trackers which only analyze the arrival of the tracked item online. E. Hinton, et al.," in [12] introduces a massive feed ahead neural network is educated on a small schooling set, it usually plays poorly on held-out test data. This over fitting is drastically decreased through randomly omitting half of the characteristic detectors on every training case. This prevents complex co-adaptations in which a characteristic detector is most effective helpful within the context of numerous different particular function detectors.

III. Problem Formulation

The main challenge in this research is to increase the performance of deep learning. To evaluate the performance, the mean absolute percentage error is calculated as an indicator for an efficient performance. If this error decreases theperformance increases for this system.Assume the variable Yˆ, are as good as possible. If Y is a categorical variable then the learning task is a classification problem. If Y is numerical variable, then studying assignment is a regression trouble. Without loss of generality, the resulting models can be as follows: A classifier or classification rule is a function $\phi : X \rightarrow Y$, where Y is a finite set of classes stand for {c1, c2, ..., cJ}.Input and output variables X1,......Xp and Y are random variables taking jointly their values from $X \times Y$ with respect to the joint probability distribution P(X, Y), where X denotes the random

vector (X1, ...,Xp). That is, P(X = x, Y = y) is the probability that random variables X and Y take values x and y from X and Y when drawing an object uniformly at random from the universe Ω. Now we search for an algorithm whose predictions are as good as possible can be stated as finding a model which minimizes its expected prediction error, defined as follows:The expected prediction error, also known as generalization error or test error, of the model ϕL is as equation 1.

$$Prederror\ (L) = X,Y\{L(Y, L(X))\} \quad (1)$$

where L is the learning set used to build ϕL and L is a loss function measuring the discrepancy between its two arguments. Equation.1 basically measures the prediction error of ϕLover all possible objects in Ω each represented by a couple $(x, y) \in X \times Y$, including the observed couples from the learning set L but also all the unseen ones from $(X \times Y) \setminus L$.

The goal is not in fact to make the very most accurate predictions over the subset L of known data, but rather to learn a model which is correct and reliable on all possible data. The most commonplace loss function is the 0-one loss functionwhere all misclassifications are equally penalized.The generalization error of L becomes the probability of misclassification of the model as depicted in equation 2.

$$Prederror\ (\mathrm{L}) = \mathrm{EX,Y}\ \{1(\mathrm{Y}= \mathrm{L(X)})\}=\mathrm{P}(\mathrm{Y}= \mathrm{L(X)})\ (2)$$

For regression, the most used loss function is the squared error loss $\mathrm{L(Y,\ L(X))} = (\mathrm{Y} - \mathrm{L(X)})^2$, where large differences among the accurate values and the predicted values are penalized more heavily than small ones. The generalization error of the model becomes as depicted in equation 3.

$$Prederror\ (\mathrm{L}) = \mathrm{EX,Y}\{(\mathrm{Y} - \mathrm{L(X)})^2\}\ (3)$$

Estimating $Prederror(\mathrm{L})$ In practice, the probability distribution P(X,Y) is usually unknown, making the direct evaluation of $Prederror(\mathrm{L})$ infeasible. It is often not possible to draw additional data, thusbuilding infeasible experimental estimation of $Prederror(\mathrm{L})$ on a set L0 drawn independently from L. In most problems, L constitutes the only data available, on which both the model needs to be learned and its generalization error estimated. The generalization error in Equation 1 can however be estimated in several ways. To make notations clearer, let us first define $Prederror(\mathrm{L,L0})$ as the average prediction error of the model L over the set L0. The first and simplest estimate of the generalization error is the reconstitution estimate or training sample estimate.

IV. Proposed Methodology

This section proposed a deep getting to improve the performance of deep learning using the proposed algorithm RFHTMC that merge the operations of both forest and HTM algorithms. There are a few center characteristics specificfrom conventional learning algorithms. feature choice: within the original dataset, no longer all capabilities are equally critical. amongst diverse capabilities, a few can be misleading to forecast set of rules, and others may additionally result in over-fitting. With those misleading capabilities and beside the point attributes inflicting over-fitting. Consequently, disposing of these redundant and irrelevant features. The dataset can enhance the functioning of the resultingforecasting set of rules. Assume the levels of random forest is denoted as LV1,.......LVn. Then the conventional prediction error in all levels is expressed as in equation (4)

$$Prederror = \sum_{i=1}^{n}\left(Prederror(\mathrm{L}) = \mathrm{X,Y}\{\mathrm{L}(\mathrm{Y},\mathrm{L(X)})\}\right)\ (4)$$

The principle of tree dependent models is superbly simple. It is composed in approximating the partition of the Bayes version via recursively partitioning the input space X into subspaces after which assign regular prediction values by 2Y to all objects x inside every terminal subspace. At this point the prediction error is expressed in equation (5) as follow:

$$Prederror = \sum_{i=1}^{n}\left(\mathrm{Err(L)} = \mathrm{EY}\{\mathrm{L}(\mathrm{Y},\mathrm{L(2Y)})\}\right)\ (5)$$

First define the subsequent concepts that merge the random forest algorithm with HTM learning algorithm.A tree is a graph G = (V, E) in which any two vertices are linked via precisely one path. A rooted tree is a tree wherein one of the nodes has been positive as the foundation. In our case, we additionally assume that a rooted tree is a directed graph, wherein all edges are directed far from the basis. If there exists an side from t1 to t2 then node t1 is stated to be the figure of node t2 while node t2 is said to be a child of node t1. In a rooted tree, a node is stated to be internal if it has one or extra children and terminal if it has no youngsters. Terminal nodes are also known as leaves.HTM divides the space Xt that node t represents into disjoint subspaces respectively similar to every of its kids, then the total prediction error$terr$ is calculated in each subspace according to equation (6).

$$terr = \sum_{i=1}^{n1}\left(Prederror(\mathrm{Li})\right) + \sum_{d=1}^{n2}\left(Prederror(\mathrm{Ld})\right)..\sum_{s=1}^{nn}\left(Prederror(\mathrm{Ls})\right)(6)$$

HTM areas clearly save transitions between sparse distributed representations. In some instances the transitions can appear to be a linear series, along with the notes in a melody, but in the well-known case many feasible future inputs may be expected on the equal time. An HTM location will make one-of-a-kind predictions primarily based on context that would stretch again far in time. most of the people of reminiscence in an HTM is dedicated to sequence reminiscence, or storing transitions among spatial patterns. The following algorithm depicts the proposed algorithm as a pseducode.

Algorithm: Random Forest HTM Cortical RFHTMC Learning Algorithm

```
Use  Random_Forest_Classifier
X (predictor) and Y (target) for training data set as an input
x_test(predictor) of test_dataset as an input
for c in active_Columns(t)
Predicted = false
For  i = 0 to cells_Per_Column - 1
If predictive_State(c, i, t-1) == true then
s = get_Active_Segment(c, i, t-1, active_State)
    if s.sequence_Segment == true then
    model= Random_Forest_Classifier()
    model.fit(X, y)
    active_State(c, i, t) = 1
if  bu_Predicted == false then
    for i = 0 to cells_Per_Column - 1
    active_State(c, i, t) = 1
for  c, i in cells
    for s in segments(c, i)
    if segment_Active(c, i, s, t) then
    predicted= model.predict(x_test)
function Predict(x)
t = t0
while t is not a terminal node do
t = the child node t0 of t
end while
return  t
end function
```

This algorithm depends on using a random forest classifier as well of assimilation the key function of HTM learning algorithm. Testing of the data sets is tested first, then the prediction process is started including the calculation of the mean absolute percentage error for all the nodes. Then going through towards the active node, Till finishing all the active cells of the nodes. The fitting model during the prediction process is applied via HTM algorithm by dviding each active path as sub active segments.

V. RESULTS AND DISCUSSION

The proposed algorithm is implemented using MATLAB software. A comparison is held between RFHTMC and HTM through two case studies. Where case Study I explains the behavior of RFHTMC based on mean absolute percentage error, mixed integer representation of input data units, permanence, overlap duty cycle and active duty cycle. The same parameters are explained in Case Study 2 that refers to the behavior of HTM algorithm. Figure 3 shows that the maximum mean absolute percentage error in the case of RFHTMC is 5% at different iteration numbers. Comparing this result with the results in figure 4, it is clear that in the case of HTM the maximum mean absolute percentage error approaches to 10%. A comparative study regarding the maximum mean absolute percentage error among RFHTMC and HTM algorithms is shown in figure 5. The proposed algorithm proves it can reduce this error by half comparing with the previous algorithm. From figure 6 the permanence at RFHTMC has more values than HTM algorithm as depicted in figure 7; this means the proposed algorithm is more durable than HTM algorithm. A comparative study regarding the stability features among the two algorithms is depicted in figure 8 that refers to RFHTMC is more stable than HTM algorithm starting from observation No. 3 till observation No. 11.

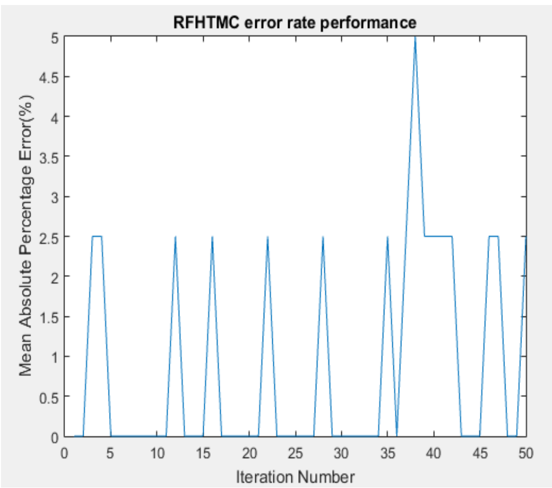

Fig.3 Mean Absolute Percentage Error at RFHTMC

Figure 9 refers to the overlap duty cycle at RFHTMC ; it is clear that the overlap duty cycle has high percentage comparing with HTM algorithm as depicted in figure 10. Practically, This an important result because, the higher the duty cycle, the longer a solenoid remains open, the more flow and less pressure develops. Figure 11 and figure 12 indicate that the mixed integer representation of input data units and the active duty cycle at RFHTMC have the same values comparing to HTM algorithm respectively.

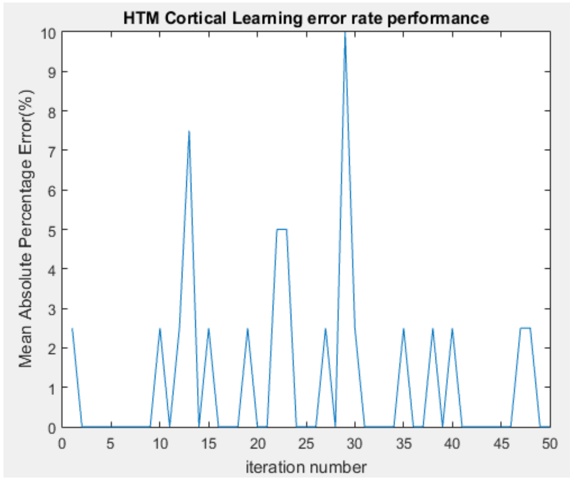

Fig.4 Mean Absolute Percentage Error at HTM

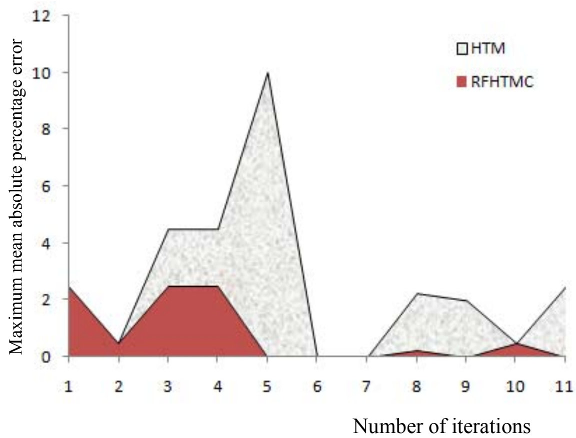

Fig.5 Maximum mean absolute percentage error comparison between RFHTMC and HTM algorithms

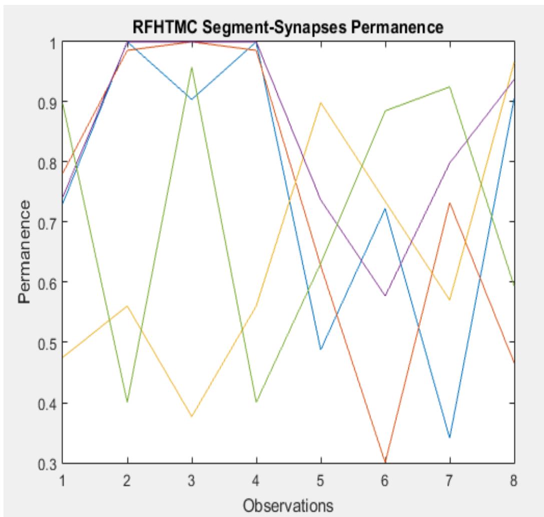

Fig.6 Permanence at RFHTMC

It is concluded from the above mentioned figures that the proposed algorithm reduces the mean absolute percentage error by half and increases the overlap duty cycle by 15%. It increase the performance of the deep learning system by the previous values and it in the same time can preserve the values of active duty cycle, the mixed integer representation of input data units and the permanence or the durability of the system.

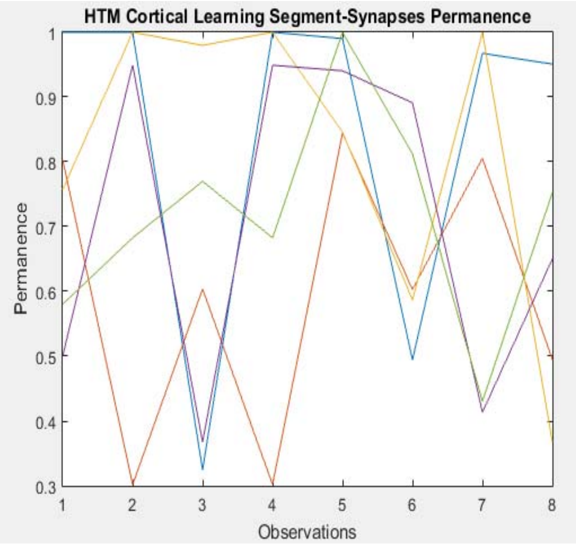

Fig.7 Permanence at HTM

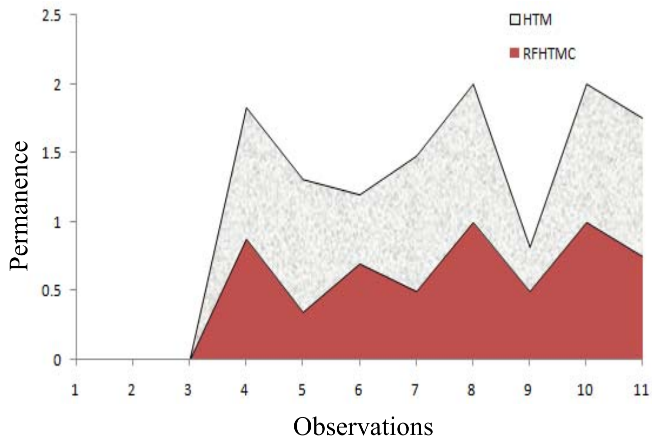

Fig.8 Permanence comparison between RFHTMC and HTM algorithms

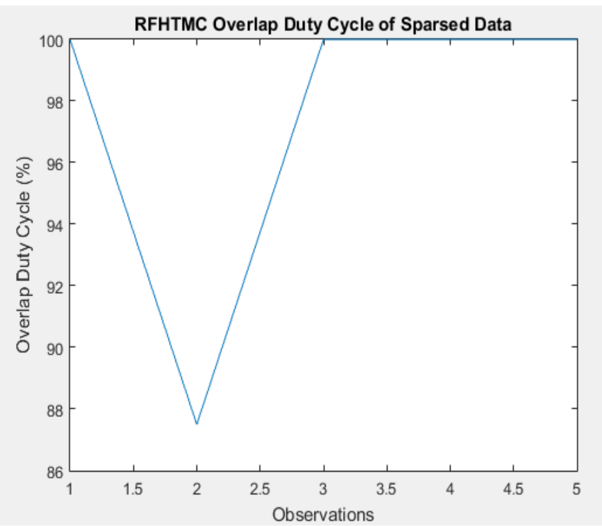

Fig.9 Overlap duty cycle at RFHTMC

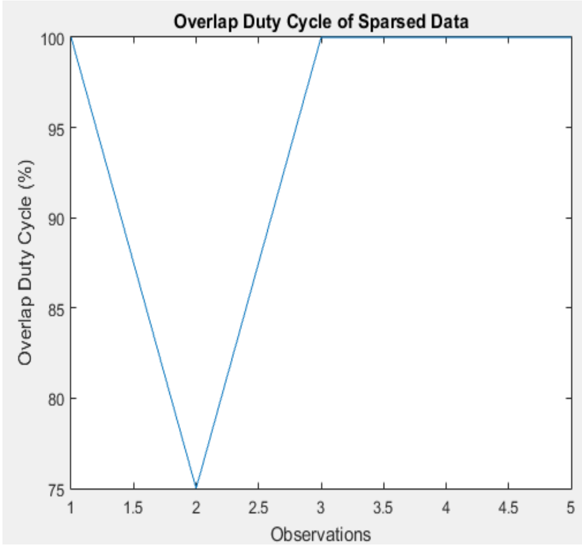

Fig.10 Overlap duty cycle at HTM

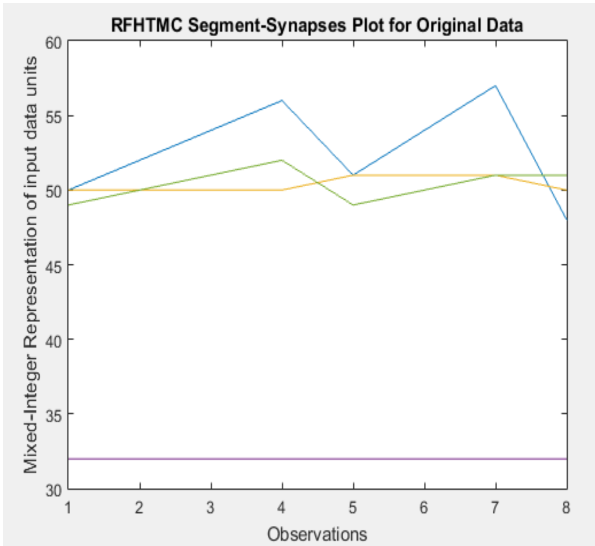

Fig.11 mixed integer representation of input data units at RFHTMC and HTM

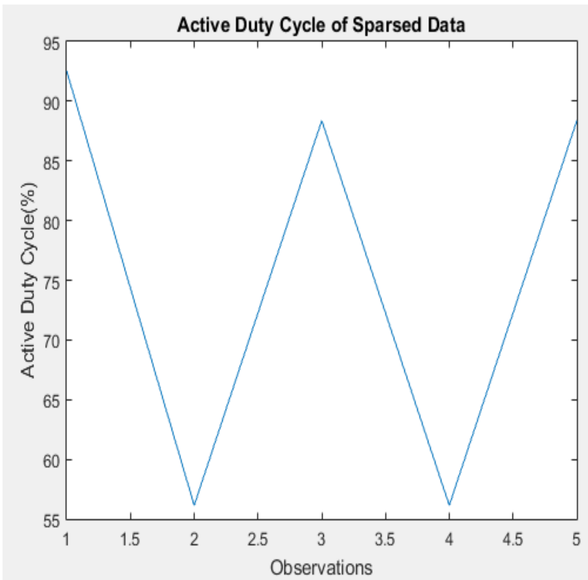

Fig.12 Active duty cycle for both RFHTMC and HTM

Conclusion

Deep learning is an important field in machine learning using Artificial Intelligence. The using of these systems depends on its behavior and performance. The main objective of this paper is to improve the performance of this type using a combined version from Random Forest and HTM Cortical Learning Algorithm. The proposed algorithm is called RFHTMC. The results depict that the proposed algorithm can reduce the mean absolute percentage error by half. And increases the overlap duty cycle by 15%.The proposed set of rules can increase the overall performance of the deep gaining knowledge of system by means of the preceding values. And it within the identical time can maintain the values of active duty cycle. Also, it maintains the mixed integer representation of input data units and the durability of the system.

References

[1] A. Alahi, okayGoel, et al. "Social LSTM: Human Trajectory Prediction in Crowded Space" the IEEE convention on computer vision and pattern recognition,2016, pages 961:971.

[2] V. Badrinarayan, et al. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling." , 2015, arXiv preprint arXiv:1505.07293.

[3] N. D. Bruce, et al. "A Deeper Look at Saliency: Feature Contrast, Semantics, and Beyond." the IEEE convention on pc vision and sample reputation, 2016, pages 516:524.

[4] Milad Malekipirbazari et al.,"Risk assessment in social lending via random forests",Expert Systems with Applications, 15 June 2015,Volume 42, Issue 10, Pages 4621:4631

[5] Jeff Hawkins et al." Sequence memory for prediction, inference and behaviour",Phil. Trans. R. Soc. B (2009) 364, 1203:1209,doi:10.1098/rstb.2008.0322

[6] Spyros Gidaris et al. "LocNet: Improving Localization Accuracy for Object Detection" the IEEE conference on laptop vision and pattern popularity, 2016, pages 789:798.

[7] D. P. Kingma et al. "ADAM: A method for stochastic optimization"In worldwide convention on studying illustration, 2015.

[8] Nian Liu et al."DHSNet: Deep Hierarchical Saliency Network for Salient Object Detection". the IEEE conference on pc imaginative and prescient and pattern recognition,2016, pages 678:686.

[9] Mostafa Mehdipour et al. "A Comprehensive Analysis of Deep Learning Based Representation for Face Recognition", CVPR Workshop on Biometrics, DOI: 10.1109/CVPRW.2016.20,At Las Vegas,USA

[10] A. Sharma, et al. "Deep hierarchical parsing for semantic segmentation". 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),DOI: 10.1109/CVPR.2015.7298651

[11] Naiyan Wang, et al. "Learning a deep compact image representation for visual tracking"Advances in Neural Information Processing Systems, Volume 26 , 2013, pages 809:817.

[12] E. Hinton, et al." Improving neural networks by preventing co-adaptation of feature detectors",2012/07/03, arXiv preprint.