Response to the Associate Editor and Reviewers

Paper: A Dynamic Neural Network Approach for Solving Nonlinear Inequalities Defined on A Graph and Its Application to Distributed, Routing-free, Range-free Localization of WSNs

Paper number: NEUCOM-D-12-00170

The authors would like to thank the reviewers and the Editor for their constructive comments.

I. RESPONSE TO ASSOCIATE EDITOR

Comment:

Three reviews of your manuscript have been received so far. Three of them suggest a minor revision and resubmit. So, based on these comments (please see the detail in the attachment), I recommend to the EIC that your paper cannot be published in current form. You may resubmit your paper after a minor revision. Please seriously and carefully answer the questions mentioned by the reviewers.

Response:

We answer the questions mentioned by the reviewers in detail as stated in the following parts.

II. RESPONSE TO REVIEWER 1

Comment:

1. the introduction is not enough. a lot of works for this area are not introduced. the relationship between the mathematical model and the localization algorithm is not well introduced and looks not smooth.

Response:

• As to works for this area, a new section is plugged in the introduction part to introduce works on recurrent neural networks for problem solving, which reads,

"To a constrained[14,16] often design a recurrent neural network evolving along the gradient descent direction of an augmented energy function formed by introducing a constraint-related penalty term into the objective function. However, the solution of the neural network often has a small deviation from the optimal one due to the compromise between the cost function and the constraints. To remedy this, [17,18] study the problem in the dual space and use a projection function to represent inequality constraints. For the case with linear inequality constraints, it is proven in theory that this type of methods are able to converge to the optimal solution. The results are latter applied to various applications, such as kinematic control of redundant manipulators [19], k-winner-take-all (k-WTA) problem solving [20], L_1 norm estimation [21], to exploit the real-time processing capability of the neural networks. Successive works [22-24] in this field extend the results to the cases with nonlinear constraints with guaranteed convergence of the recurrent neural network. Although great success has been gained in the field of using recurrent neural networks to solve constrained optimization, difficulties are encountered when using recurrent neural networks to solve nonlinear inequalities defined on a large scale graph as the recurrent neural network approach is essentially a centralize method and is not scalable to a large networked systems. By taking advantage of the local connection between nodes on a graph, a distributed recurrent neural network, whose topology is isomorphic to the graph topology, is elaborately designed to solve nonlinear inequalities defined on the graph in a decentralized fashion. "

- On the second issue about the mathematical model and the localization algorithm, the problem concerned in this paper is formulated as nonlinear inequalities defined on a graph as given in (1). This formulation is an abstraction of real problems in networked environments, e.g., communication connectivity maintenance of a robot network, relationship or friendship construction in social networks, range-free localization of wireless sensor networks. Particularly, we apply the proposed method to range-free localization of wireless sensor networks in Section 4 as an application example to demonstrate the application potentials of the proposed method. Clearly, the problem of range-free localization defined in (15) falls into the general problem (1) by choosing $f_{ij}(x_i, x_j) = (x_i - x_j)^T(x_i - x_j) - R^2$ in (1). To clarify this relation between problem (1) and the range-free localization of wireless sensor networks and smooth the transition, we add Remark 3 to highlight this point. Remark 3 reads,

"**Remark 3** By choosing $f_{ij}(x_i, x_j) = (x_i - x_j)^T(x_i - x_j) - R^2$ in (1), the problem of range-free localization defined in (15) falls into the unified framework given in problem (1) and thereby can be efficiently solved using the proposed recurrent neural network approach. Other applications with networked constraints can be solved in a similar procedure if they fall into the framework presented in (1)."

Comment:

2. in the simulation part, the DOI IS NOT used. we cannot see the feasible of the proposed algorithm. also, pls compare the proposed algorithm with other range-free algorithms.

Response:

The major contribution of this paper is to propose a neural network approach for solving inequalities defined on a graph. To demonstrate the potentials of the proposed approach in real applications, we apply the approach to range-free localization of WSNs. The concentration of our paper is not the range-free localization, and therefore the comparisons with other range-free algorithms are not conducted. We thank the reviewer to point out this to us and we will take this into account in our future work concentrating on range-free localization of WSNs. The codes used in the simulations are uploaded to http://personal.stevens.edu/∼lshuai/codes/.

Comment:

3. some of the recent works for this area should be cited, H. Chen, P. Deng, Y. Xu and X. Li, A Novel Localization Scheme Based on RSS Data for Wireless Sensor Networks, Proc. APWeb 2006 International Workshop on Sensor Networks, LNCS 3842, pp. 315-320, 2006

Q. Shi, C. He, H. Chen, and L. Jiang, "Distributed Wireless Sensor Network Localization via Sequential Greedy Optimization Algorithm," IEEE Transactio ns on Signal Processing, vol. 58, no. 6, pp. 3328-3340, Jun. 2010.

H. Chen, M.H.T.Martins, P.Huang, H.C.So, and K.Sezaki, "Cooperative node localization for mobile sensor networks," Proc. 2008 International Conference On Embedded and Ubiquitous Computing (EUC 2008), vol.I, pp.302-308, December 2008, Shanghai, China

M.H.T.Martins, H. Chen, and K.Sezaki, "OTMCL: Orientation Tracking-based Monte Carlo Localization for Mobile Sensor Networks," Proc. of the Six International Conference on Networked Sensing Systems (INSS'09), USA

H. Chen, Q. Shi, R. Tan, H. V. Poor, and K. Sezaki, "Mobile Element Assisted Cooperative Localization for Wireless Sensor Networks with obstacles," IEEE Transactions on Wireless Communications, vol. 9, no. 3, pp. 956-963, Mar. 2010.

Response:

Thank you for recommending the related works, which improve our understanding of the state-of-the-art in this field. we have added them into the reference list.

Comment:

4. the complexity of the proposed algorithm is not clear since the neural network is with high

complexity. pls give the complexity analysis.

Response:

- Time Complexity

Due to the parallelism nature of the neural network, all parts of the neural circuit work simultaneously in parallel, the computation complexity is relative low although there exists nonlinear elements in the circuit. Note that each neural module associated with a sensor node. It only interacts with its neighbors. The introduction of extra sensor nodes outside the neighborhood has no direct impact on this particular neural module. Consequently, the convergence of the neural dynamics of a particular neural module is independent of the total number of sensors in the network (instead, it has dependance on the number of neighbors, i.e., the degree of a node on the graph in the terminology of graph theory). Therefore, the time complexity is $O(1)$.

- Spatial Complexity

As to the spatial complexity, which is often concerned in the neural network community, we can find that each module (as shown in Fig. 1) consists of a limit number of analog devices and therefore the spatial complexity is $O(n)$ (n denoting the number of nodes in the sensor network), meaning that the total consumption of analog devices is linear dependent on the number of sensor nodes in the network. We summarize this point in the newly added remark, which reads,

"**Remark 4** Each neural module associates with a sensor node. It only interacts with its neighbors. The introduction of extra sensor nodes outside the neighborhood has no direct impact on this particular neural module. Consequently, the convergence of the neural dynamics of a particular neural module is independent of the total number of sensors in the network (instead, it has dependance on the number of neighbors, i.e., the degree of a node on the graph in the terminology of graph theory). Therefore, the time complexity of the proposed approach is $O(1)$ for its circuit implementation. As to the spatial complexity, we can find that each neural module (as shown in Fig. 1) consists of a limit number of analog devices and therefore the spatial complexity is $O(n)$ (n denotes the number of nodes in the sensor network), meaning that the total consumption of analog devices is linear dependent on the number of sensor nodes in the network."

III. RESPONSE TO REVIEWER 2

Comment:

In this paper, the authors studied how to get a feasible solution satisfying a set of inequalities defined on a graph. By using the KKT conditions, they proposed a recurrent neural network approach to get a

solution of the problem and proved its feasibility and convergence. Then they applied their approach to range-free localization problems and demonstrated the results through simulations.

Detailed comments are listed below: 1) Is the solution given by the recurrent neural network globally optimal? Satisfying the KKT condition only guarantee that a solution is locally minimal. In the proof of Lemma 1 in page 6, the authors declared that the equilibrium point is an optimal solution to the problem. Does "optimal" mean globally optimal or locally?

Response:

The solution given by the recurrent neural network is globally optimal. To show this, we only need to demonstrate that the constrained optimization problem (2) is convex, i.e., both the objective function and the constraints are convex, as the KKT condition gives the globally optimal solution for convex optimization problems. For problem (2), the objective function is a constant and therefore is convex. As to the constraints, each single constraint inequality $f_{ij}(x_i, x_j) \leq 0$ is a convex set since the nonlinear function $f_{ij}(x_i, x_j)$ is convex as stated in the text below the inequality (1) (also as stated in Theorem 1). All inequalities together for all possible i and j forms the intersection of all these convex sets and therefore is also convex. Thus, problem (2) is indeed convex. Its optimal solution is identical to the solution of the KKT condition. Specifically, for inequality (15), which is the constraint used in the application, it is convex as mentioned in the proof of Corollary 1. Accordingly, the result referred in Corollary 1 is also globally optimal.

Comment:

2) With the recurrent neural network approach, it seems that neighboring nodes need to exchange some information such as x_i to update their state. This will incur additional communication overhead, which is related to the iterative times of the neural network. Please add some analysis on this.

Response:

As claimed in Remark 1, communication only happens between neighbor nodes for information exchange. As such, no cross-hop communication exists in the network and the communication protocol can be thoroughly tailored to save time for communication. For the case without taking communication time into account, the iterative time is purely dependent on the neural network structure and can be reduced by increasing the value of the scaling factor ϵ. For the case that the communication time is not negligible, the communication time introduces delay in the dynamics of the neural network as the received information from neighbors is the version τ seconds ago (say τ is the communication time) instead of the current one. In such a situation, the convergence conditions can be obtained by solving linear matrix inequalities as done in [30]. Coarsely speaking, the total iterative time in the case with non-negligible

communication time can be approximately measured by the summation of the communication time for the required number of iterations and the computation time taken by the neural network. We summarize this point in the new added remark Remark 2, which writes,

"**Remark 2** For the case with non-negligible communication time, the communication time introduces delay in the dynamics of the neural network as the received information from neighbors is the version τ seconds ago (say τ is the communication time in seconds) instead of the current one. In such a situation, the convergence conditions can be obtained by solving linear matrix inequalities as done in [30]. Coarsely speaking, the total iterative time in the case with non-negligible communication time can be approximately measured by the summation of the communication time for the required number of iterations and the computation time taken by the neural network."

Comment:

3) What's the purpose of Fig. 4, Fig. 5, Fig. 8, and Fig. 9? It would be clearer how the proposed algorithm performs if the authors summarize the average localization error of their approach.

Response:

The equilibrium point of the proposed recurrent neural network is the optimal solution of the constrained optimization problem. In order to reach the equilibrium point ultimately, the neural network must be convergent. Theorem 1 proves the convergence of the neural network in theory. We use Fig. 4, Fig. 5, Fig. 8, and Fig. 9 to show the neural network indeed converges with time in simulation. As shown in the figures, with random initial values, the state values converge to constants after a short period of transient, which in turn validates the theoretical conclusion on convergence. As suggested by the reviewer, we performed simulations and added a new paragraph in Section 6.2 to summarize the average localization error. The new added part related to this point is as follows:

We use $E = \sqrt{\sum_{i=1}^{n} (x_i - x_i^r)^T (x_i - x_i^r)/n}$ with x_i^r denoting the real position of the ith blind sensor node, to evaluate the localization accuracy of the proposed method. Table I shows the localization error and the simulation time averaged by running Monte Carlo simulation for 50 times. The simulation is performed with the programming language Matlab 7.8 on a laptop with the Intel (R) Core(TM) 2 Duo CPU at 1.80 GHz and 2GB of RAM. Note that the simulation program performs the localization algorithms for all the beacon nodes and all the blind nodes. In real application, the localization algorithm will be run in a distributed manner separately by all nodes. In all simulations, the beacon nodes are evenly deployed in space. As observed in the table, the localization error is around 0.15 for the case with 9 beacons. With the increase of beacons, the localization error decreases. For the case with 36 beacons, the changes of blind node numbers have small influences to the localization error and the value is around 0.6. As to the

Parameters		Performances		
No. of beacons	No. of blind sensors	Average Localization Error	Average PC time(seconds)	Theoretical time (seconds)
9	110	0.1756	11.193993	1.0×10^{-4}
9	130	0.1701	15.187071	1.0×10^{-4}
9	150	0.1309	20.810211	1.0×10^{-4}
9	170	0.1549	26.104286	1.0×10^{-4}
16	110	0.0995	14.219097	1.0×10^{-4}
16	130	0.0904	17.293275	1.0×10^{-4}
16	150	0.0912	23.444137	1.0×10^{-4}
16	170	0.0947	27.156888	1.0×10^{-4}
25	110	0.0751	15.964024	1.0×10^{-4}
25	130	0.0700	19.236726	1.0×10^{-4}
25	150	0.0669	24.460955	1.0×10^{-4}
25	170	0.0684	31.058264	1.0×10^{-4}
36	110	0.0672	16.791616	1.0×10^{-4}
36	130	0.0658	21.397013	1.0×10^{-4}
36	150	0.0645	26.174775	1.0×10^{-4}
36	170	0.0619	31.232622	1.0×10^{-4}

TABLE I

PERFORMANCES UNDER DIFFERENT PARAMETER SETUPS.

PC time (the CPU time in the simulation), it increases with the increase of the number of sensor nodes. It is worth noting that the theoretical running time of the proposed algorithm is much less than the PC time. This is because, on one hand, the simulation program simulates all nodes on a single computer while the algorithm is expected to run separately on all sensors in parallel in real application and on the other hand the neural network implementation in analog circuits can complete the computation when the neural evolution converges. As to the software implementation of the neural network model, the running time can be estimated by the ratio between the PC time listed in the table and the total number of nodes simulated, whose value is still acceptable for real-time processing.

Comment:

4) Some typos: Page 2, the last paragraph, there are two "Section 4", but "Section 5" and "Section 6" are not mentioned. Page 3, above Eq. (4), "(11)" is not given in the paper Page 5, the second line below Eq. (7): "iV" possibly should be "$i \in V$" In many sentences, "such as" and "etc." are both used; only one should be used.

Response:

We thank the reviewer for pointing out the typos to us. It really helps us to improve the quality of this

paper. We have revised the typos in the new version.

IV. RESPONSE TO REVIEWER 4

Comment:

The convergence and feasibility of the neural network are investigated in this paper. Some applications of the proposed method are also applied to solve the range-free localization problem in WSNs. (1)The English presentation needs further improvement and the authors should carefully check and correct them in the revision for the whole paper, i.e., in page 2, line 5, in the last paragraph, "to solving"

Response:

Thank you for pointing out our presentation problems. We have corrected them in our revised version.

Comment:

(2)The abstract should be simplified and focus more on the proposed method and contribution of this paper. Remark 1 should also be simplified.

Response:

In the revised version, the abstract part is simplified into the following,

"In this paper, we are concerned with the problem of finding a feasible solution to a class of nonlinear inequalities defined on a graph. A recurrent neural network is proposed to tackle this problem. The convergence of the neural network and the solution feasibility to the defined problem are both theoretically proven. The proposed neural network features a parallel computing mechanism and a distributed topology isomorphic to the corresponding graph. Thus it is suitable for distributed real-time computation. The proposed neural network is applied to range-free localization of wireless sensor networks (WSNs). The analog circuit implementation of the neural network for such an application is also explored. Simulations demonstrate the effectiveness of the proposed method."

To illustrate the distributed property of the proposed neural network model clearly and summarize the main point in the remark briefly, we move the detailed analysis in the original version of Remark 1 outside and state the main point in the revised version of remark 1. The related parts read,

"In the dynamic equation (6), the update of x_i, which is associated with the vertex i, requires information on the value of x_i itself, the value of x_j for $j \in \mathbb{N}(i)$, which is associated with i's neighbor vertex j, the value of λ_{ii} associated with the vertex i, the value of λ_{ij} associated with the edge $\{i, j\} \in \mathbb{E}$ and the value of λ_{ji} associated with the edge $\{j, i\} \in \mathbb{E}$. All the above required information comes either from the vertex i itself or from its neighbor vertices or the edges in between. On the other hand, for the update of λ_{ij}, which is associated with the edge $\{i, j\} \in \mathbb{E}$, the required information includes the value

of λ_{ij} itself, the value of x_i and the value of x_j, which are associated with the two vertices forming the edge $\{i, j\}$. Clearly, all the required information for the update of λ_{ij} is available in the neighborhood. In this sense, the neural network (6) has an isomorphic topology to the graph $\mathcal{G}(\mathbb{V}, \mathbb{E})$.

Remark 1 The connection topology of the neural network model (6) is isomorphic to the graph $\mathcal{G}(\mathbb{V}, \mathbb{E})$, on which the problem (1) is defined. This property enables us to implement the neural network in a distributed fashion by assigning the state variables x_i, λ_{ii} and λ_{ij} for all $j \in \mathbb{N}(i)$ to the vertex i. In this way, for the dynamic evolution of the neural network (6), communications only happen between neighbor vertices and neither routing nor cross-hop communication is required, which thoroughly reduces the communication burden especially for the case with a large number of vertices."

Comment:

(3)The main contribution of this paper is Theorem 1 in Section 3. From the proof of this theorem, I have some doubts that what's the specific expression of the solution of the network, and what the decay rate of the solution. The expression of "exponentially converge." Is not clear for the readers.

Response:

The proposed neural network is described by the dynamic equation (6). Its equilibrium point is the solution of the optimization problem. The equilibrium point satisfy equation (10). However, the nonlinear equation (10) cannot be analytically solved and therefore we cannot obtain the specific expression of the solution. To get the numerical solution, we use the proposed neural network to reach the solution recursively.

The conclusion on "exponentially convergence" of the neural network is obtained based on Lemma 2. For the proposed neural network, exponentially convergence means that the difference between the neural state and the desired equilibrium point is always within a bounded region defined by a function with a constant convergence rate decaying to zero (i.e., the function is an exponential function with respect to time). Similar to related existing works [29], Theorem 1 in this paper also does not convey information on the value of the decay rate of the neural network but the property of exponentially convergence grantees that the decay rate is lower bounded. Actually, the time scale of the neural network can be adjusted by tuning the scaling factor ϵ. By increasing the value of ϵ, the decay rate can be increased.

Comment:

(4)The signification of each simulation should be given in more detail. In line 2 before Section 7, "The value starts from 387.1349 and drops to 0.0106", but I can't find these data on the simulation graph.

Response:

The data can be read from Fig. 10 (note that the y-axis in this figure is in logarithmic scale). From this

figure, we can see that the initial value at time $t = 0$ is between 10^2 and 10^3 (actually the accurate value is 387.1349) and the value at the end of this simulation is around 10^{-2} (the accurate value is 0.0106).

Comment:

(5)The novelty of the results relative to previous works should be explained and the motivation on the study should be further emphasized.

Response:

This paper is concerned with solving nonlinear inequalities defined on a graph. A prominent feature of the proposed neural network is that its topology is isomorphic to the graph topology, i.e., each neuron in the neural network corresponds to a node on the graph and the neural connections corresponds to edges on the graph, which is in contrast to the conventional neural networks with connections between all pairs of neurons. For applications, this topological feature enables distributed computation, which is often necessary for a scalable large scale network. To stress this point as the major novelty of the result and to stress the desire of distributed computation on a graph in applications, which motivates our study, this section of description is inserted into the introduction part of this paper.