A load balancing scheme based on deep-learning in IoT

Hye-Young Kim[1] · Jong-Min Kim[2]

Received: 15 July 2016 / Revised: 11 October 2016 / Accepted: 14 October 2016 / Published online: 17 November 2016

Abstract Extending the current Internet with interconnected objects and devices and their virtual representation has been a growing trend in recent years. The Internet of Things (IoT) contribution is in the increased value of information created by the number of interconnections among things and the transformation of the processed information into knowledge for the benefit of society. Benefit due to the service controlled by communication between objects is now being increased by people who use these services in real life. The sensors are deployed to monitor one or more events in an unattended environment. A large number of the event data will be generated over a period of time in IoT. Hence, the load balancing protocol is critical considerations in the design of IoT. Therefore, we propose an agent Loadbot that measures network load and process structural configuration by analyzing a large amount of user data and network load, and applying Deep Learning's Deep Belief Network method in order to achieve efficient load balancing in IoT. Also, we propose an agent Balancebot that processes a neural load prediction algorithm based on Deep Learning's Q-learning method and neural prior ensemble. We address the key functions for our proposed scheme and simulate the efficiency of our proposed scheme using mathematical analysis.

✉ Hye-Young Kim
hykim@hongik.ac.kr

✉ Jong-Min Kim
dyuo1004@gmail.com

[1] Major in Game Software, School of Games, Hongik University, Sejong Campus, Sinan-ri, Jochiwon-eup, Sejong, Korea

[2] Department of Convergence Security, Kyonggi University, Iui-dong, Yeongtong-gu, Suwon-si, Gyeonggi, South Korea

Keywords Load balancing · Internet of things · Deep belief network · Q-learning

1 Introduction

The term Internet of Things (IoT) was first coined by Kevin Ashton in 1999 in the context of supply chain management [1]. However, in the past decade, the definition has been more inclusive covering wide range of applications like healthcare, utilities, transport, etc. [2]. Although the definition of 'Things' has changed as technology evolved, the main goal of making a computer sense information without the aid of human intervention remains the same. The IoT could allow people and things to be connected anytime, anyplace, with anything and anyone and any services. This is stated as well in the ITU vision of the IoT, according to "From anytime, anyplace connectivity for anyone, we will now have connectivity for anything" [3].

The vision of future internet based on standard communication protocols considers of computer network IoT, Internet of People (IoP), Internet of Energy (IoE), Internet of Media (IoM) and Service of Internet (IoS) into a global IT platform of seamless network. This will create new opportunities in a large of aspect such as a smart health, retail, green energy, manufacturing, smart home, smart city and personal application. Benefit due to the services controlled by communication between objects is now being increased by people who use these services in real life [4]. The IoT contribution is in the increased value of information created by the number of interconnections among things and the transformation of the processed information into knowledge for the benefit of society.

The quantity of data on the Internet is growing and it is around 2.5 quintillion bytes of data and it is estimated that 90 % of the data was generated in the past two years [5].

The sensors are deployed to monitor one or more events in an unattended environment. A large number of the event data will be generated over a period of time in IoT. Hence, the load balancing protocol is critical considerations in the design of IoT.

Therefore, we propose an agent Loadbot that measures network load and process structural configuration by analyzing a large amount of user data and network load, and applying Deep Learning's Deep Belief Network method in order to achieve efficient load balancing in IoT. Also, we propose an agent Balancebot that processes a neural load prediction algorithm based on Deep Learning's Q-learning method and neural prior ensemble. We address the key functions for our proposed scheme and simulate the efficiency of our proposed scheme using mathematical analysis.

The rest of the paper is structured as follows. Section 2 includes the background introduction about open load balancing and Deep Learning issues. In Sect. 3, we describes our proposed architecture and provide the detailed a load balancing scheme which is proposed an agent Loadbot and Balancebot that process a measures network load and process structural configuration and a neural load prediction algorithm and balance of the load in order to achieve efficient load balancing in IoT. In the final section, we constitute a summary of our proposal and suggest a further study directions networks.

2 Related works

The Internet of Things is a novel paradigm that is rapidly gaining ground in the scenario of modern wireless telecommunications. The basic idea of this concept is the pervasive presence around us of a variety of things or objects—such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, and mobile phones which through unique addressing schemes are able to interact with each other and cooperate with their neighbors to reach common goals [6]. With the widespread deployment of networked, intelligent sensor technologies, an Internet of Things (IoT) is steadily evolving, much like the Internet decades ago. One Example of the widespread IoT technologies is the Google's Nest Thermostat, the famous IoT gadget during 2014. Their designers disclosed that it can become carbon neutral in a period of just that were created by manufacturing and distributing the device are offset by the energy savings one obtains from using it [7].

In the literature we have found several IoT architectures such as LinkSmart Project [8], RestThing [9], S3OiA [10], Gao [11], Wang [12] and Weiss [13].

As with the growth of the Internet of Things market, the number of autonomous devices with wireless connection capabilities is expected to grow significantly. As opposed to human generated traffic, the traffic is mostly based on reporting sensor measurements on the uplink. Although such traffic is not expected to create congestion at the backbone due to small message sizes, it may result in a congestion during the channel access phase if a large number of M2M devices access the channel in a short time frame. Such a congestion is considered as a serious problem because it may result in significant access delays [14]. There have been numerous proposals for the load control of the random access channel [15–20]. These proposals are mostly based on announcing an access probability, where the nodes are barred from channel access with some probability. Service differentiation provided by these methods is limited: they either treat all nodes equally by announcing the same access probability; or, they group nodes into few predetermined service classes as in Extended Access Barring [21] and treat them according to their access class. Such a limited service differentiation is not enough for satisfying the wide range of service requirements of IoT devices. Without load balancing, a coordinator with excessive number of wireless sensor nodes may lead to more energy consumption and experiences data reception delay as compared to the other coordinators.

Recent breakthroughs in computer vision and speech recognition have relied on efficiently training deep neural networks on very large training sets. The most successful approaches are trained directly from the raw inputs, using lightweight updates based on stochastic gradient descent. By feeding sufficient data into deep neural networks, it is often possible to learn better representations than handcrafted features [22]. Perhaps the best-known success story of reinforcement learning is TD-gammon which learnt entirely by reinforcement learning and self-play, and achieved a superhuman level of play [23].

However, early attempts to follow up on TD-gammon, including applications of the same method to Chess, Go and Checkers were less successful. This led to a widespread belief that the TD-gammon approach was a special case that only worked in backgammon, perhaps because the stochasticity in the dice rolls helps explore the state space and also makes the value function particularly smooth [24].

The sensors are deployed to monitor one or more events in an unattended environment. A large number of the event data will be generated over a period of time in IoT. Hence, the load balancing protocol is critical considerations in the design of IoT.

Therefore, we propose an agent Loadbot that measures network load and process structural configuration by analyzing a large amount of user data and network load, and applying Deep Learning's Deep Belief Network method in order to achieve efficient load-balancing in IoT. Also, we

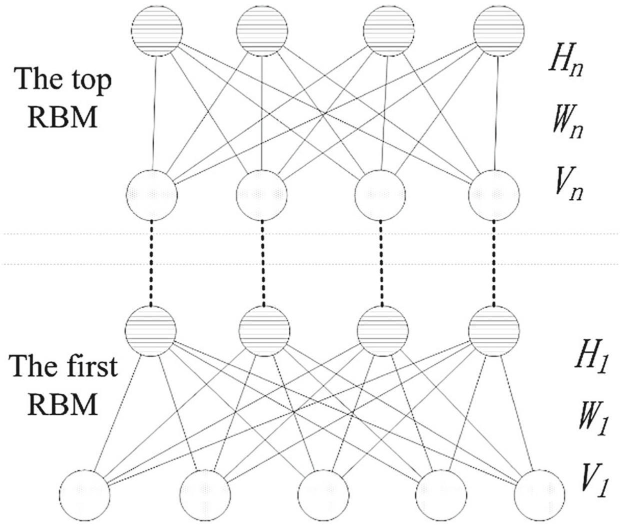

Fig. 1 Deep belief network [29]

propose an agent Balancebot that processes a neural load prediction algorithm based on Deep Learning's Q-learning method and neural prior ensemble.

3 Proposed scheme

3.1 Drawing out structure map of network load

In order to model complicate nonlinear relationship such as IoT network, we drew out the structure map of network load by using Deep Belief Network which is graph generating model and connected to each layer by in-depth connection between multiple layers which created nerve network [22, 25–28].

In other words, in Deep Belief Network which is for modeling distinguished objects, each objects can be represented as hierarchical organization of basic image element, extra layers can muster features from low ranked layers, and model Network load from complicate data in a greedy way with smaller number of nodes in Fig. 1 [29]. By applying Deep Belief Network, we drew out the Network structure map in IoT environment as following.

$$P\left(x, h^1, \ldots, h^l\right) = \left(\prod_{k=0}^{l-2} P\left(h^k | h^{k+1}\right)\right) P\left(h^{l-1}, h^l\right) \tag{1}$$

① First layer is an input of $x = h^0$ and is taught by Restricted Boltzmann Machine (RBM). $h$means layer.

② We make second layer by using the first layer

③ We also train the second layer by RBM.

$$\Delta w_{ij}(t+1) = w_{ij}(t) + \eta \frac{\partial \log(p(v))}{\partial w_{ij}} \tag{2}$$

④ Repeat second and third procedure in order to meet the requirement. Below is the explanation of simple algorithm by using Pseudo Code

```
numpy_rng = numpy.random.RandomState(123)
print’... building the model’ # construct the Deep Belief
Network
dbn = DBN (numpy_rng=numpy_rng,
           n_ins=28*28,
           hidden_layers_sizes=[1000,1000,1000],
           n_outs=10)
```

3.2 Network load learning

Reinforcement Learning of Deep Learning is acquired through action that can be taken while carrying out network load learning in a given environment and reinforcement value in a scalar form which we take from an assessment on a chosen action. This method can be effectively applied in network load learning because it can be performed in dynamic environment effectively. This learning requires dynamic learning process of filtering variables depending on network circumstances and applying it in order to weigh the input value.

Q-learning approach does not calculate optimum action from current state but learn through optimum action on each condition based on experience of number of trial and error.

Therefore, it can make decision on each action based on learned policy without complicated calculation which cost time. It can also decide optimum action based on compensation value from each environment even though it may not be aware of compensation value or probability value of every environment beforehand. Using existing input and output data set, it learn to output wanted material when input is inserted. For network load decision algorithm, using Q-learning, which is one of the model-free reinforcement learning technique, to calculate and store current network load and learn its outcome value and it also enable ability to predict the outcome of network load by long-term analysis on massive data. Q value is set to 0 or initialized to a random number. Arrow shows possible actions in each states. Assuming that 100 weight value is given only when action pointed toward G is performed and rest gets none, update rule about Q value cause weight value of target to slowly spread to all possible state which enable us to make Q-table about optimum network load. Model-free reinforcement learning is a technique to let autonomy agent, which sense and calculate network load, to select optimum value to fulfill its purpose. This approach can effectively applied to myopically optimal load decision strategy which does not have forecasting ability. Therefore, this research plans to use following algorithm

Table 1 Parameters of modelling

Parameters	Descriptions
S	Number of server in IoT environment
U	Number of users
p_u	Users' transmit and receive rate
p_s	Servers' transmit and receive rate
C_s	Service load value of server
V_u	Users' weighted value
w_x	User centered load value
w_y	Suggested load value by Loadbot
L_b	User 'b's load value
E_s	Effectiveness of load balancing of server S

for the research. Variables used in modeling algorithm are as follows in Table 1.

In IoT environment, there exist S number of serves and B number of users. It is assumed that server, at random time, adjust p_s by existing dynamic load balancing and user 'b' request transmit receive service, load value of user b, L_b is following:

$$L_b\left(p_s\right) = \begin{cases} v_b - p_s & \text{if } \mathrm{p}_4 \leq v_6 \\ 0 & otherwise \end{cases} \tag{3}$$

We assume w_x when applying user centered load value and w_y when applying load value suggested by Loadbot. Then variable showing effectiveness of load balancing, E_s is,$E_s\left(p\right) = \left(p_s - c_s\right) D_s\left(p\right)$ and $D_s\left(p\right)$ is user request rate provided by server. Therefore, $D\left(p\right) = \rho U h_s\left(p\right) g D_s\left(p_s\right)$ and $h_s\left(p\right)$ signify percentage of server to be chosen by user, $g\left(p_s\right)$ is $v_u \geq p_s$ which means user ratio. Therefore, Eq. (4) means calculation of user centered load value,

$$h_s\left(p\right) = w_x \int_s^x \left(p\right) + w_x \int_s^x \left(p\right), \int_s^x \left(p\right) \tag{4}$$

$w_x \int_s^x \left(p\right)$ means calculation of load value by Loadbot.

$$E_s\left(p\right) = \left(p_s - c_s\right) h_s\left(p\right) g\left(p_s\right) \tag{5}$$

In IoT network, for network load balancing, learn *E: S-> A*. Cumulative reward accumulated by random E is

$$\begin{aligned} V\left(s,\pi\right) &= \textstyle\sum_{t=0}^{\infty} \gamma E\left(r_t | \pi, s_0 = s\right) \\ V\left(s,\pi\right) &= r\left(s, a_\pi\right) + \gamma \sum V\left(s', \pi\right) \end{aligned} \tag{6}$$

In here, s_0 is initial state, r_t is a weight value ins time t. a_π is decided by Loadbot, s' shows the next state. By fundamental concept of Q-learning [30],

$$Q\left(s,a\right) = r\left(s,a\right) + r \sum_{s'} V\left(s', u^*\right) \tag{7}$$

Next shows Q-learning approach and in this research, we apply this algorithm to apply it on network load learning.

```
For each state s and action a, randomly initialize table element;
  Perceive current states
  Do forever {
        Select action a and carry it out;
        Get immediate reward r;
        Perceive new state s'; renew table element about
          Q(s,a) (Q: estimate value about real Q);
        Q (s, a) ← r + $\gamma$ maxa' Q(s', a')
        s ← s';
  }
```

We model value of $w_x \int_s^x \left(p\right)$ and calculate $Q\left(s,a\right)$ to materialize Loadbot that measure load in IoT.

3.3 Load balancing using neural prior ensemble

When new data is entered, it is stacked to certain amount and made into new Deep Belief Network and stored. At this moment, new Deep Belief Network receive previous network information. Later in this paper, Neural Prior Ensemble [31] combine and learn from all network load and draw out the process of modifying weighted value. Based on the analysis of collection of massive data, we made new data value by approximating relationship between data. Next shows Neural Prior Ensemble approach which will used in this paper to Balancebot.

```
  Do forever {
        Save new data  $D_{new}$ in the network
         $W_{prev}$, a new deep belief network, initialize  $W_{new}$
        Learn new data value  $D_{new}$
        $\theta$   < $-\theta$  U $W_{new}$
        Get immediate reward r
        Identify new circumstance s'; renew table element
        about Q(s, a) (Q: estimate value about real Q);
     Q (s, a) ← r + $\gamma$maxa' Q(s', a')
     s ← s';
  }
```

In this paper, in order to acquire network load balancing, Q-Learning approach from Reinforcement Learning and Neural Prior Ensemble are applied to suggest new network load learning technique. When new data is entered, it is stacked to certain amount and made into new Deep Belief Network and stored. At this moment, new Deep Belief Network receive previous network information. We combine the Neural Prior Ensemble with our Loadbot to learn from all network load and draw out the process of modifying weighted value.

Therefore, we propose an agent Balancebot that processes a neural load prediction algorithm based on Deep Learning's

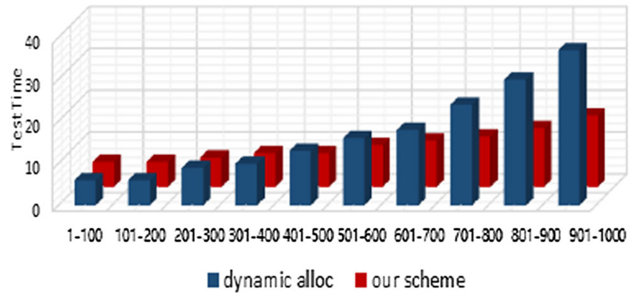

Fig. 2 Migration by number of nodes

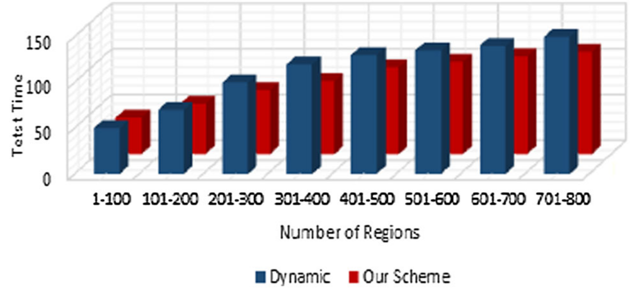

Fig. 3 Migration by number of regions

Q-learning method and neural prior ensemble as in the above scheme.

3.4 Experimental result

In this section, we introduce the perform simulation in IoT to evaluate the effectiveness and performance of our approach. Then we compare and analyze the result of our scheme and others. MATLAB10 was used in our simulation for a performance analysis.

The load problem being defined as a nonlinear problem with linear constraints, we also present the results obtained by the simulation. We obtain the proportions of traffic very close to the optimal case. In Figs. 2 and 3, we note that the most consuming is proportional when the number of sensor nodes is very large and network regions in IoT.

Our scheme is better than the dynamic scheme even though the number of sensor nodes increased, the occurrence of migration for load-balancing did not increase as much in IoT. As shown in Fig. 2, when the number of mobile nodes exceeds 700, our scheme is a difference between the dynamic schemes.

In addition, we triggered messages between mobile nodes in order to measure the migration of regions in IoT. We note that such migration occurs less frequently than dynamic scheme as Fig. 3.

4 Conclusion

In this paper, we have proposed a load balance scheme through Loadbot and Balancebot that measure network load and process structural configuration by analyzing a large amount of user data and network load, and applying Deep Learning's Deep Belief Network method and processes a neural load prediction algorithm based on Deep Learning's Q-learning method and neural prior ensemble in order to achieve efficient load-balancing in IoT. We address the key functions for our proposed scheme and simulate the efficiency of our proposed scheme using mathematical modelling. Then, we compare the results with the previous researches. Our results show the effeteness of ours, and it is required that further works for more comparison with previous researches.

Acknowledgements This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. 2016RIA2B4012386) and this work was supported by 2016 Hongik University Research Fund.

References

1. Ashton, K.: That "Internet of Things". RFiD J. **22**, 97 (2009)
2. Sundmaeker, H., Guillemin, P., Friess, P., Woelfflé, S.: Vision and Challenge for Realising the Internet of Things. Cluster of European Research Projects on the Intern of Things-CERP IoT (2010)
3. ITU Internet Reports, the Internet of Things (November 2005)
4. Atzoti, L., Iera, A., Morabito, G.: The Internet of Things: a survey. Comput. Netw. **54**, 2787–2805 (2010)
5. IBM, Bringing Big Data to the Enterprise. http://www-01.ibm.com/software/data/bigdata/. Accessed 10 July 2012
6. Giusto, D., Iera, A., Morabito, G., Atzori, L. (eds.): The Internet of Things. Springer, New York (2010). ISBN: 978-1-4419-1673-0
7. http://net.com/blog/2014/04/22todays-earth-day-tomorrow-shold-be-too
8. Qin, W., et al.: RestThing: a Restful Web service infrastructure for mash-up physical and Web resources. In: Proceedings of the EUC 2011, 197–204 (2011)
9. Vega-Barbas, M., Casado-Mansilla, D., et al.: Smart spaces and smart objects interoperability architecture (S3OiA). In: Proceedings of the IMIS 2012, 725–730 (2012)
10. Gao, L., Zhang, C., et al.: RESTful Web of Things API in sharing sensor data. In: Proceedings of the ICITST 2011, 1–4 (2011)
11. Wang, H.-I.: Constructing the Green Campus within the Internet of Things architecture. J. Distrib. Sens. Netw. 1–8 (2014)
12. Weiss, M., Guinard, D.: Increasing energy awareness through web-enabled power outlets. In: Proceedings of the MUM 2010, 20–30 (2010)
13. López-de-Armentia, J., Casado-Mansilla, D., López-de-Ipiña, D.: Reducing energy waste through eco-aware every-day things. J. MIS **10**(1), 79 (2014)
14. Hasan, M., Hossain, E., Niyato, D.: Random access for machine-to-machine communication in LTE-advanced networks: issues and approaches. IEEE Commun. Mag. **51**(6), 86–93 (2013)
15. CATT, Access control of mtc devices (r2-100182) Technical Report (2010)
16. Phuyal, U., Koc, A., Fong, M.-H., Vannithamby, R.: Controlling access overload and signaling congestion in M2M networks. Asilomar conference on signals, systems and computers, 2012, pp. 591–595

17. Galinina, O., Turlikov, A., Andreev, S., Koucheryavy, Y.: Stabilizing multi-channel slotted aloha for machine-type Communications. In: IEEE ISIT, pp. 2119–2123 (2013)
18. Duan, S., Shah-Mansouri, V., Wong, V.: Dynamic access class barring for M2M communications in LTE networks. IEEE global communications conference (GLOBECOM), pp. 4747–4752 (2013)
19. Wu, H., Zhu, C., La, R., Liu, X., Zhang, Y.: FASA: accelerated S-ALOHA using access history for event-driven M2M communications. IEEE/ACM Trans. Netw. **21**(6), 1904–1917 (2013)
20. Cheng, M.Y., Lin, G.Y., Wei, H.Y., Hsu, A.C.: Overload control for machine-type-communications in LTE-Advanced system. IEEE Commun. Mag. **50**(6), 38–45 (2012)
21. 3GPP, Study on RAN improvements for machine-type communications (release 11), Technical Report (2011)
22. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1106–1114 (2012)
23. Barto, A., Konidaris, G.: Applying deep belief networks to the game of Go. In: MIT Press (May 2010)
24. Artaro, L., Perez, Q.: A model for general video game learning. Desseration for Master Degree of Universidad Nacional De Colombia (May 2016)
25. O'Connor, P., Neil, D., Liu, S.C., Delbruck, T., Pfeiffer, M.: Real-time classification and sensor fusion with a spiking. Front. Neurosci. vol. 17, pp. 61–73, October 8, 2013. Article ID 178
26. Salama, M.A., Hassanien, A.E., Fahmy, A.A.: Deep belief network for clustering and classification of a continuous data. The 10th IEEE international symposium on signal processing and information technology, pp. 472–477, Dec 2010
27. Maei, H., Szepesv'ari, C., Bhatnagar, S., Sutton, R.S.: Toward off-policy learning control with function approximation. In: Proceedings of the 27th international conference on machine learning (ICML 2010), pp. 719–726 (2010)
28. Yu, L., Yang, Z., Tang, L.: A novel mutistage deep belief network based extreame learing machine ensemble learning paradigm for credit risk assessment. In: Flexible Services and Manufacturing Journal, pp. 1–17, September 29, 2015
29. Liu, F., Liu, B., Sun, C., Liu, M., Wang, X.: Mutimodal deep belief network based link prediction and user comment generation. In: ICONIP 2015, pp. 20–28
30. van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double Q-Learning. In: Proceedings of the thirtieth AAAI conference on artificial intelligence (2015)
31. Wang, X.L., Chen, Y.Y., Zhao, H., Lu, B.L.: Parallelized extreme learning machine ensemble based on min–max modular network. In: Neurocompting, vol. 128, pp. 31–41, March 29, 2014

Hye-Young Kim received the B.Sc. degree and M.S. degree in Computer Science from the SookMyung University in South Korea and Ph.D. degree in Computer Science and Engineering from the Korea University of South Korea in February 2005. During her Ph.D. study, she had focusing on location management scheme, such as mobile IPv6, cellular network and network mobility. Currently, she works at Hongik University of South Korea as an Associate Professor since March 2007. She had developed a network protocol for 9 years (1989–1997) while she was working at Hyundai Electronics as a senior researcher. She had taught in the Department of Computer Game at Chung Kang College as an assistant professor for three years (2001–2004). Her research interests include traffic modeling and location management in internet of thing and wireless network for mobile game or multimedia service.

Jong-Min Kim received the Ph.D. degree in Industrial Security from the Kyonggi University, Korea in 2015. He is currently professor of Convergence Security Department, Kyonggi University, Korea. His research interests are Industrial Security, Malware prediction, Convergence Security and information security.