Computer Science & Information Technology

Natarajan Meghanathan Jan Zizka (Eds)

Computer Science & Information Technology

6th International Conference on Advanced Information Technologies and Applications (ICAITA 2017) October 28~29, 2017, Sydney, Australia



AIRCC Publishing Corporation

Volume Editors

Natarajan Meghanathan, Jackson State University, USA E-mail: nmeghanathan@jsums.edu

Jan Zizka, Mendel University in Brno, Czech Republic E-mail: zizka.jan@gmail.com

ISSN: 2231 - 5403 ISBN: 978-1-921987-74-8 DOI: 10.5121/csit.2017.71301 - 10.5121/csit.2017.71305

This work is subject to copyright. All rights are reserved, whether whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the International Copyright Law and permission for use must always be obtained from Academy & Industry Research Collaboration Center. Violations are liable to prosecution under the International Copyright Law.

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

The 6th International Conference on Advanced Information Technologies and Applications (ICAITA 2017) was held in Sydney, Australia, during October 28~29, 2017. The 6th International Conference on Soft Computing, Artificial Intelligence and Applications (SAI 2017), and 4th International Conference on Signal and Image Processing (Signal 2017) was collocated with The 6th International Conference on Advanced Information Technologies and Applications (ICAITA 2017). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The ICAITA-2017, SAI-2017, Signal-2017 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically. All these efforts undertaken by the Organizing and Technical Committees led to an exciting, rich and a high quality technical conference program, which featured high-impact presentations for all attendees to enjoy, appreciate and expand their expertise in the latest developments in computer network and communications research.

In closing, ICAITA-2017, SAI-2017, Signal-2017 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the ICAITA-2017, SAI-2017, Signal-2017.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

Natarajan Meghanathan Jan Zizka

Organization

General Chair

David C. Wyld	Southeastern Louisisna University, USA
Jan Zizka	Mendel University in Brno, Czech Republic

Program Committee Members

Ajao Lukman Adewale Ali Dorri Ali Zaart Amani Samha Amine Laghrib Amir Rastegarnia Ankit Chaudhary Apai, Asadullah Shaikh Assem Mousa Atheer Yousif Oudah Bai Li **Barbaros** Preveze Barhoumi Walid Braham Barkat Carlo Sau Chaker LARABI Chandan Kumar Karmakar Chuanzong Zhang Dac-Nhuong Le Dalel BOUSLIMI Duan Keqing Fabio Gasparetti Hacer Yalim Keles Hamid Ali Abed AL-Asadi Hanming Fang Hassan Ugail Issac Niwas Swamidoss Klimis Ntalianis Lark Kwon Choi Mohammadreza Balouchestani Mohammed Fatehy Soliman Mokhtar Mohammadi Naveed Ahmed Petrellis N Raad Ahmed Hadi Ramgopal Kashyap

Federal University of Technology, Nigeria Islamic Azad University, Iran Beirut Arab University, Lebanon Amani samha researcher QUT, Australia Faculté des Sciences Beni-Mellal, Marocco University of Malayer, Iran Truman State University, USA Universiti Malaysia Perlis, Malaysia Najran University, Saudi Arabia E commerce tech support systems manager, Egypt Thi-Qar University, Iraq Woodside Energy Ltd, Australia Cankaya University, Turkey SIIVA-LIMTIC Laboratory, ENICarthage, Tunisia The petroleum Institute, Saudi Arabia Università degli Studi di Cagliari, Italy Université de Poitiers, France University of Melbourne, Australia Aalborg University, Denmark Haiphong University, Vietnam Institut Mines- Telecom, France Wuhan Early Warning Academy, China Roma Tre University, Italy Ankara University, Turkey Basra University, Iraq Logistical Engineering University, China University of Bradford, UK Nanyang Technological University, Singapore Athens University of Applied Sciences, Greece The University of Texas at Austin, USA Indiana Purdue Fort Wayne University, USA Suez Canal University, Egypt Shahrood University of Technology, Iran University of Sharjah, UAE TEI of Thessaly, Greece Iraqi University, Iraq SISTec. India

Salem Nasri Samuel Baraldi Mafra Supriya Karmakar Touahni. Raja, Upasna Vishnoi Wee kuok kwee Yuanchang Sun Yueying Kao Yuying Shi Zili Li Selahattin KOSUNALP Shah J Miah Son Nguyen Thai Sos Agaian Tanmoy Sarkar Tanzila Saba Tonghan Wang Tzung-Pei Hong Wan Shuai Xianglei Xing

Qassim University, Saudi Arabia Federal University of Parana (UFPR), Brazil Mellanox Technologies, USA IBN Tofail University, Morocco Sr. Digital Design Engineer, USA Multimedia University, Malaysia Florida International University, USA Chinese Academy of Sciences, China North China Electric Power University, China Delft University of Technology, Netherland University of Bayburt, Turkey Victoria University, Australia Tra Vinh University, Vietnam The University of Texas at San Antonio, USA Microsoft Corporation, USA Prince Sultan University, Kingdom of Saudi Arabia East China University of Technology, China National University of Kaohsiung, Taiwan Northwestern Polytechnical University, China Harbin Engineering University, China.

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)



Information Technology Management Community (ITMC)



Organized By



Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

6th International Conference on Advanced Information Technologies and Applications (ICAITA 2017)

Accelerated Bayesian Optimization for Deep Learning	01 -	- 13
Ayahiko Niimi and Kousuke Sakamoto		

6th International Conference on Soft Computing, Artificial Intelligence and Applications (SAI 2017)

Optimal Sizing and Placement of Distributed Generators and Capacitors in		
Radial Distribution Network	27 - 3	7
Partha P Biswas, P. N. Suganthan and Gehan A J Amaratunga		

4th International Conference on Signal and Image Processing (Signal 2017)

ACCELERATED BAYESIAN OPTIMIZATION FOR DEEP LEARNING

Ayahiko Niimi¹ and Kousuke Sakamoto²

 ¹Faculty of Systems Information Science, Future University Hakodate, 2-116 Kamedanakano, Hakodate, Hokkaido 041-8655, Japan
 ²School of Systems Information Science, Future University Hakodate, 2-116 Kamedanakano, Hakodate, Hokkaido 041-8655, Japan

ABSTRACT

Bayesian optimization for deep learning has extensive execution time because it involves several calculations and parameters. To solve this problem, this study aims at accelerating the execution time by focusing on the output of the activation function that is strongly related to accuracy. We developed a technique to accelerate the execution time by stopping the learning model so that the activation function of the first and second layers would become zero. Two experiments were conducted to confirm the effectiveness of the proposed method. First, we implemented the proposed technique and compared its execution time with that of Bayesian optimization. We successfully accelerated the execution time of Bayesian optimization for deep learning. Second, we attempted to apply the proposed method for credit card transaction data. From these experiments, it was confirmed that the purpose of our study was achieved. In particular, we concluded that the proposed method can accelerate the execution time when deep learning is applied to an extremely large amount of data.

KEYWORDS

Deep Learning, Bayesian Optimization, Activation Function, Real Dataset

1. INTRODUCTION

In this paper, we propose a deep-learning automatic parameter-tuning method using an improved Bayesian optimization.

Deep learning is a new approach, which has recently attracted considerable attention in the field of machine learning. It considerably improves the accuracy of abstract representations by reconstructing deep structures, such as the neural circuitry of the human brain. Moreover, deep learning algorithms have been honored in various competitions, such as the International Conference on Representation Learning.

However, a problem with deep learning is that its performance cannot be optimized unless multiple parameters are appropriately tuned. Manual tuning of the parameters is performed based on the experience and intuition of experts. To address this problem, many studies have investigated automatic parameter tuning. Furthermore, it has been proposed that the learning algorithm's parameters can be determined by automatic parameter tuning. However, this requires the original machine-learning algorithm to be repeated multiple times. If applied to a heavy learning model, such as deep learning, this process becomes time-consuming.

Therefore, in this study, we propose a deep-learning automatic parameter tuning method using an improved Bayesian optimization, which is one of the methods used in automatic parameter-tuning algorithms. The proposed method aims at accelerating the execution time of Bayesian optimization for deep learning. This method focuses on the relation between output values and the accuracy of each neuron. The output value of each neuron is obtained by substituting the total input into the activation function (Relu). If this value is 0, then the feature value is also 0, indicating that the neuron could not successfully extract the feature. In other words, the probability that learning does not proceed increases as the number of 0's in the output value of each neuron increases. From this fact, when learning is performed using this method, the number of 0s in the output value of each neuron excessively increases and learning stops. By performing learning using the following parameters, the execution time of Bayesian optimization is increased.

The novelty of this research is that Bayesian optimization is applied to deep learning. Bayesian optimization has been applied to light learning models, but there have been no studies applied to heavy learning models like deep learning. In this research, we explore the possibility of deep learning using Bayesian optimization through experiments with actual credit card data.

This study is summarized as follows: first, in section 2 and 3, we describe deep learning and Bayesian optimization. In section 4, we propose the improved Bayesian optimization for deep learning. In section 5 and 6, we describe the experiment and results of the study, respectively. Finally, in section 7, we discuss our conclusions and future work.

2. DEEP LEARNING

Deep learning is a new approach, which has recently attracted considerable attention in the field of machine learning.

Deep learning is a generic term for multilayer neural networks that have been studied over several years [1]-[3]. Multilayer neural networks reduce the overall calculation time by performing calculations on hidden layers. Therefore, these networks are prone to excessive overtraining because an intermediate layer is often used to approximate each layer.

Nevertheless, technological advancements have addressed the overtraining problem, while the use of GPU computing and parallel processing has increased the tractable number of hidden layers.

A sigmoid or tanh function has been commonly used as the activation function (Equations (1) and (2)). However, recently, the maxout function has also been used (section 2.1) and the dropout technique has been implemented to prevent overtraining (section 2.2).

$$h_i(x) = \operatorname{sigmoid}(x^T W_{\dots i} + b_i) \tag{1}$$

$$h_i(x) = \tanh(x^T W_{\dots i} + b_i) \tag{2}$$

2.1. Maxout

The maxout model is simply a feed-forward architecture such as a multilayer perceptron or deep convolutional neural network that uses a new type of activation function, the maxout unit [3].

In particular, given an input $x \in \mathbb{R}^d$ (x may be either v or a hidden layer's state), a maxout hidden layer implements the function

$$h_i(x) = \max_{j \in [1,k]} z_{ij} \tag{3}$$

where $z_{ij} = x^T W_{\dots ij} + b_{ij}$, $W \in \mathbb{R}^{d \times m \times k}$ and $b \in \mathbb{R}^{m \times k}$ are learned parameters. In a convolutional network, a maxout feature map can be constructed by taking the maximum across \$k\$ affine feature maps (i.e., pool across channels, in addition to spatial locations). When training with the dropout, in all cases, we perform element-wise multiplication with the dropout mask immediately prior to multiplication by weights and the inputs not dropped to the max operator. A single maxout unit can be interpreted as a piecewise linear approximation of an arbitrary convex function. In addition to learning the relationship between hidden units, maxout networks also learn the activation function of each hidden unit.

The maxout approach abandons many of the mainstays of traditional activation function design. Even though the gradient is highly spare, the representation produced by maxout is not sparse at all, and the dropout will artificially sparsify the effective representation during training. Although the maxout unit may saturate on one side or another, this is a measure zero event (so it is almost never bounded from above). Because a significant proportion of the parameter space corresponds to the function delimited from below, maxout learning is not constrained. Moreover, the maxout function is locally linear almost everywhere, in contrast to many popular activation functions that demonstrate significant curvature. Considering all these deviations from thestandard practice, it may seem surprising for the maxout activation functions to work however, we find that they are very robust, easy to train with dropout, and achieve excellent performance.

$$h_i(x) = \max_{j \in [1,k]} Z_{ij} \tag{4}$$

$$z_{ij} = X^T W_{\cdots ij} + b_{ij} \tag{5}$$

2.2. Dropout

Dropout is a technique that can be applied to deterministic feedforward architectures that predict an output γ given an input vector v [3].

In particular, these architectures contain a series of hidden layers $h = \{h(1), \dots, h(L)\}$. Dropout trains an ensemble of models comprising a subset of the variables in both v and h. The same set of parameters θ are used to parameterize a family of distributions $p(y \mid v; \theta, \mu)$, where $\mu \in \mathcal{M}$ is a binary mask determining which variables to include in the model. For each example, we train a different submodel by following the gradient $\log p(y \mid v; \theta, \mu)$ for a different randomly sampled μ . For many parameterizations of p (usually for multilayer perceptrons) the instantiation of the different submodels $p(y \mid v; \theta, \mu)$ can be obtained by element-wise multiplication of v and h with the mask μ .

The functional form becomes important when the ensemble makes a prediction by averaging the submodels' predictions. Previous studies of bagging averages used the arithmetic mean. However, this is not possible with the exponentially large number of models trained by dropout. yield can models easily Fortunately, some а geometric mean. When $p(y \mid v; \theta) = \operatorname{softmax}(v^T W + b)$, the predictive distribution defined by renormalizing the geometric mean of $p(y \mid v; \theta, \mu)$ over \mathcal{M} is simply given by softmax $(v^T W/2 + b)$. In other words, the average exponential prediction for many submodels can be computed simply by running the full model with the weights divided by two. This result holds exactly in the case of a single layer softmax model. Previous work on dropout applies the same scheme to deeper architectures, such as multilayer perceptrons, where the W/2 method provides only an

approximation of the geometric mean. While this approximation is not mathematically justified, it performed well in practice.

3. BAYESIAN OPTIMIZATION

The automatic parameter-tuning method automatically determines the machine-learning parameters. Parameters are considerably important in machine learning, and the accuracy significantly varies depending on how the parameters are set. However, a great amount of knowledge and experience is required to determine appropriate parameters. Automatic parameter tuning can complement knowledge and experience and make machine learning easier to handle.

Grid search is one of the automatic parameter-tuning techniques. In this technique, multiple parameter combinations are made to determine the parameter with the best accuracy. However, the problem with grid search is that there is a possibility that some parameter combinations which were not tested may actually be the best parameters. However, if the number of combinations of the tested parameters is increased, the execution time will be prolonged.

Another automatic parameter-tuning technique is Bayesian optimization. In this technique, several parameters are tested and the parameter combinations are determined by predicting the accuracy of parameter combinations that have not been tested. This method determines the parameters by predicting a combination of multiple parameters and a parameter having a high probability of being the most accurate one based on multiple parameters and their precision.

- 1) Predict the prior distribution of parameter accuracy in the Gaussian process.
- 2) Predict the posterior distribution from prior distribution by Bayesian estimation.
- 3) Search the combinations of parameters with the highest accuracy from posterior distribution using the Markov chain Monte Carlo (MCMC) method.
- 4) Repeat steps 1)–3) to select the most accurate parameter.

In the Gaussian process, the prior distribution was assumed to be a normal distribution. From the obtained data, the probability that data is obtained under certain parameters was defined as likelihood. The Gaussian process uses the fact that the likelihood approaches the posterior distribution. To find parameters with high probability of high accuracy from the accuracy of multiple models, there are many methods. In this paper, we use Metropolis-Hastings (MH) method.

In such a procedure, it is possible to predict a combination of parameters with the highest accuracy and high probability. The problem in Bayesian optimization is that the combination of the predicted parameters may not be the best parameter.

This method is a method for predicting accuracy and parameter prior distribution. The method is performed in the following procedure. This method predicts the accuracy and parameter prior distribution. It has the following procedure:

- 1) Find the accuracy of multiple parameters.
- 2) Assume that the accuracy of the obtained parameters follows the normal distribution and predict the prior distribution.

4

By implementing this procedure, prior distribution can be predicted.

Bayesian estimation is one of the methods used for Bayesian optimization. This method predicts the accuracy and posterior distribution of parameters. In this method, the posterior distribution is obtained using the Bayes' theorem from the accuracy of the prior distribution, which is obtained using the Gaussian process and multiple parameter combinations. Using Bayesian estimation, the posterior distribution can be predicted.

The MCMC method is one of the methods used for Bayesian optimization [4]. This method searches a parameter combination with the highest accuracy in the posterior distribution. This method is a combination of the Markov chain and Monte Carlo method, each of which is explained as follows:

- Markov chain: The next state is determined by the previous state; however, it is not affected by the previous state.
- Monte Carlo method: This method generates random numbers from various probability distributions.

The MCMC method involves the use of various methods, such as the Guinness sampling and Metropolis–Hastings (MH) method. The MH method is generally used in this study and has the following procedure:

- 1) Determine the transition destination of the point where multiple parameters obtained by the Gaussian process are combined.
- 2) If the transition destination point has a higher probability of higher accuracy than the transition point before the transition, the transition is rejected with a certain probability when the probability of high accuracy is low. The reason for rejection is that it can enter a different mountain by transitioning to a point with lower precision and can get off the mountain of the local solution.

By repeating such a procedure, it is possible to search parameter combinations with a high probability of high accuracy in the posterior distribution.

The difference between grid search and Bayesian optimization is whether or not to consider parameters that have not been tested. In grid search, parameters are tested at regular intervals; however, because there are parameters that are not tested, it is possible that parameters with the best accuracy exist among these parameters. Bayesian optimization compensates for the defects of grid search in order to estimate the parameters with the highest accuracy among all the parameters from the accuracy of multiple and select parameters. However, as the estimated parameter accuracy is not necessarily the highest, there is a possibility of choosing a wrong parameter if the trial count is small.

This study deals with one of the methods used to accelerate Bayesian optimization by aiming to speed up the MCMC method by locally creating an approximate posterior distribution [5]. The MCMC method creates a posterior distribution from multiple parameters and precision values, determines the next transition destination, and obtains the accuracy of the transition destination parameter through the model; however, this operation is time-consuming. Therefore, speeding it up will also speed up the operation of the MCMC method.

4. PROPOSED METHOD FOR ACCELERATING THE MCMC METHOD

In this research, the proposed method aims to accelerate the execution time of Bayesian optimization for deep learning. It focuses on the relation between output values and the accuracy of each neuron. The output value of each neuron is obtained by substituting the total input into the activation function (Relu). If this value is 0, the feature value is also 0, indicating that the neuron was not successful in extracting the feature. In other words, the probability that learning does not proceed increases as the number of 0s in the output value of each neuron increases. Consequently, when learning is performed, the number of 0's in the output value of each neuron excessively increases and learning is stopped. By performing learning with the following parameters, the execution time of Bayesian optimization is increased.

The feature of the proposed method is to assess when learning becomes excessive and then stop it. Normal Bayesian optimization follows the procedure summarized below:

- 1) Find the accuracy of the deep-learning model with randomly determined parameters.
- 2) Find parameters with high probability of high accuracy from the accuracy of multiple models.
- 3) Calculate the accuracy of the deep-learning model with the obtained parameters.
- 4) Accurately compute parameters with high accuracy by adding the obtained parameters and accuracy.
- 5) Repeat steps 3) and 4) multiple times.
- 6) Select the parameter with the highest accuracy.

In this method, low-accuracy learning is performed for a number of learning times, which is timeconsuming. The proposed method solved this problem and follows the procedure summarized below:

- 1) Find the accuracy of the deep-learning model with randomly determined parameters
- 2) Find parameters with high probability of high accuracy from the accuracy of multiple models.
- 3) Calculate the accuracy of the deep-learning model with the obtained parameters. At this time, the output of the first and second layers of the model is obtained; learning which has not advanced is stopped.
- 4) Accurately compute parameters with high accuracy by adding the obtained parameters and accuracy.
- 5) Repeat steps 3) and 4) multiple times.

6) Select the parameter with the highest accuracy.

At the point where step 3) is different from normal Bayesian optimization, by stopping learning that has not advanced by performing this procedure, the proposed method can increase the execution time.

We used Python as the implementation language. The reason for using Python is because it can use TensorFlow and because it contains libraries that allow fast calculation of NumPy arrays.

In addition, a program for deep-learning modeling called TensorFlow was used because it allows calculating only the necessary parts when obtaining the output value of the neuron necessary for implementing the proposed method. This is because the speed can be increased.

Implementation was performed according to the following procedure.

- 1) Create a program to develop deep-learning models for deep learning.
- 2) Prepare a program to calculate the number of neuron output value 0 from the deep-learning model.
- 3) Create a program that can stop the model to learn from the number of neuron's 0 output values.
- 4) Create a Bayesian optimization program.
- 5) Combine a program that can stop the model to learn from the number of neuron's 0 output values and a program of Bayesian optimization.

Development was conducted in the following environment:

OS	Memory	# of CPU	Python	TensorFlow
Amazon Linux	1GB	1	Anaconda	0.11.0
			3-4.2.0	

Table 1. The Development Environment

TensorFlow is an open-source software library for numerical calculation using a data flow graph. As it was developed for machine-learning applications and deep-learning research, it comes with many functions essential for deep learning. TensorFlow was utilized due to its suitability to the intended purpose of this study [6].

TensorFlow can speed up calculation time by performing only the necessary calculations. We will explain the procedure when we want to obtain the output y(x) of the intermediate layer of the three-layer neural network, as shown in Fig.1. Calculations are made according to the following procedure.

- 1) First, define the number of neurons in each layer, weight (w), bias (b), weight adjustment method (e.g., gradient descent method), activation function, and calculation formulae.
- 2) Assign training data and real values, such as weight \$w\$, to the definition.
- 3) We used weight (w), bias (b), and input (x) from the output y(x) calculation formula.
- 4) Construct the minimum necessary model from the judged contents. In other words, create a model with layers other than the output layer.

By design, TensorFlow makes it possible to assess which calculations are essential and perform those calculations only.



Figure 1. Auto Encoder

5. EXPERIMENT 1: IRIS DATASETS

To verify the effectiveness of the proposed method, we used the Iris dataset to conduct an experiment [7]. The dataset contained 3 classes of 50 instances each, where each class referred to a type of an iris plant. (Table 2)

	Table 2.	The	Iris	Dataset
--	----------	-----	------	---------

# of data	# of training	# of test	# of attributes	# of classes
150	120	30	4	3

Bayesian optimization and the proposed method were performed 5 times. The parameters determined by Bayesian optimization are the Neuron number (1 - 30), the learning coefficient of the gradient descent method (0.1, 0.2, and 0.3), and the number of learning (2 - 1001). The experimental results are summarized in Tables 3 and 5.

Table 3. The Result of Bayes Oprimization (Iris)

# of learning	# of neurons	Accuracy	Calculation time(s)
28206	772	0.9833	153.7447
27202	782	0.9833	154.5802
27248	797	0.9833	153.6453
26004	796	0.9833	151.6539
27259	805	0.9833	154.4179

# of learning (Original)	# of learning (Actual)	# of learning (Reduction)	# of Models (Low Accuracy)	# of total nurons	Accuracy	Calculation time(s)
27316	16256	11060	17	831	0.9916	145.9402
27212	16195	11017	21	795	0.9833	145.6388
29067	23083	5984	10	760	0.9833	150.9196
27035	19124	7911	13	820	0.9916	145.0893
26590	18446	8144	13	822	0.9833	146.4700

Table 4. Result of the Proposed Method (Iris)

Based on the results, a speed increase of approximately 9s without loss in precision was observed. However, because the dataset and the decrease in the time needed were small, we considered testing our method using a larger dataset; therefore, we conducted experiments with credit card transaction data.

6. EXPERIMENT 2: CREDIT CARD TRANSACTION DATASET

The datasets for credit card transactions are as follows:

- 1) Credit approval dataset
- 2) Card transaction dataset

6.1. Credit approval dataset

For each user submitting a credit card application, a record of the decision to issue the card or reject the application is maintained. This is based on user attributes in accordance with general usage trend models.

However, to reach this decision, it is necessary to combine multiple models, each referring to a different clustered group of users.

6.2. Credit card transaction data

In actual credit card transactions, the data is complex, constantly changing, and continuously arriving online as follows:

- i. Data of approximately one million transactions arrive daily.
- ii. Each transaction takes less than 1 s to complete.
- iii. Approximately 100 transactions are done per second during peak time.
- iv. Transaction data arrive continuously.

Therefore, it is accurate to consider credit card transaction data as a stream. However, even if we use data mining for such data, an operator can only monitor approximately 2,000 transactions per day. Thus, suspicious transaction data should be effectively detected by analyzing less than 0.02% of the total number of transactions. In addition, fraud detection from the analysis of massive amounts of transaction data is extremely low because real fraud occurs at an extremely low rate, i.e., within a range of 0.02% - 0.05% of the amount of all analyzed transaction data.

In a previous study, the transaction data in CSV format were described as being attributed to a time order [8]. Credit card transaction data have 124 attributes, of which 84 are transactional data, including an attribute used to indicate fraudulent activity. The remaining data are behavioral data and are relevant to credit card usage. The inflow file size is approximately 700 MB per month.

Mining the credit card transaction data stream involves difficulties as it requires performing efficient calculations on an unlimited data stream with limited computing resources; therefore, many stream-mining methods seek an approximate or probabilistic solution instead of an exact one. However, as actual unauthorized credit card use is very less, these imprecise solutions do not appropriately detect frauds.

We are currently researching how to apply credit card data to ordinary deep learning [9], [10].

We apply the method proposed in this paper to credit card transaction data. As experiments cannot be conducted at universities due to security issues, we collaborate with non-academic researchers.

6.3. Credit card transaction data (Random Dataset)

In this study, the proposed method was also validated using a large credit card transaction dataset. We constructed this dataset from the actual credit card transaction dataset, which contained 129 attributes with random values within the same range specified for each attribute. The dataset contained approximately 32,000 transactions, including approximately 218 instances of illegal usage. While pre-processing deleted attributes, including null value and character string attributes, the number of attributes changed from 129 to 64 attributes. Because this dataset contained random values, it cannot be used to evaluate accuracy. Instead, experiments were conducted to confirm whether the calculation time could be reduced using the proposed method.

The percentage of fraudulent transactions in the dataset was considerably low. In the experiment, we used all illegal activity occurrences (218 instances) and a sample of normal usage activity (218 instances). The results of the experiments are summarized in Tables 5 and 6.

# of learning	# of neurons	Accuracy	Calculation time(s)
14942	1526	0.5481	208.9575

Table 5. Result of Bayes Oprimization (Credit Card)

# of learning (Original)	# of learning (Actual)	# of learning (Reduction)	# of Models (Low Accuracy)	# of total nurons	Accuracy	Calculation time(s)
16174	11657	4517	7	1568	0.5298	188.7191

Table 6. Result of the Proposed Method (Credit Card)

Based on the experimental results, our method could speed up the execution time by approximately 20s while maintaining the same precision. This result showed that it is highly probable that the proposed method can speed up the execution time if the amount of data is large.

6.4. Credit card transaction data (Real Dataset)

In this paper, we apply the proposed method for real transaction dataset from real system.

Because all the data is enormous, sampled data was used. The use samplinged dataset as follow (see in Table 7)

 Table 7. Credit Transaction Dataset

Number of Instances:	120,000
Number of Attributes	125
Number of Instance for Training:	20,000
Class Distribution for Training:	illegal : 382 (1.91%), legal : otherwise.
Number of Instance for Test:	100,000
Class Distribution for Test:	illegal : 1148 (1.148%), legal : otherwise.

Verification was conducted while changing the number of sampling.

In the experiment, we use the following environment.

- i. OS Linux (VM on Windows7 64bit)
- ii. CPU Intel i303229 3.30 GHz
- iii. Memory 4GB
- iv. Disk 500GB

We used the following indexes for evaluation.

- i. Correct answer rate: Among the transactions deemed to be illegal, the ratio of transactions that were illegal
- ii. Detection rate: The ratio of transactions judged to be invalid among all illegal transactions

Table 8 shows the experimental results.

As for the tuning (sampling 1), since there was a date to distinguish all test data as illegal, the result rate was low. For tuning (sampling 2), the detection rate was lower than Deep Learning alone, but the accuracy rate slightly improved. Since the number of trials is small, it is necessary to repeatedly verify in order to obtain a general-purpose result.

	Deep Learning (10 times learning)	Deep Learning (20 times learning)	Auto-Tuing (sampling 1)	Auto-Tuning (sampling 2)
# of Illegal Judgment	2986	1862	21887	1473
# of True Illegal in Judgement	567	574	608	466
Correct Answer Rate	18.99%	30.83%	2.78%	31.63%
Detection Rate	49.39%	50.00%	52.96%	40.59%

Table 8. Result of the Proposed Method (Real Credit Card)

7. CONCLUSIONS

In this paper, we proposed a deep-learning automatic parameter-tuning method with improved Bayesian optimization.

Deep learning is a new approach, which recently attracted considerable attention in the field of machine learning. However, a problem with deep learning is that it cannot demonstrate satisfactory performance unless multiple parameters are appropriately tuned. Bayesian optimization for deep learning is time-consuming because it involves several calculations and parameters. To solve this problem, this study aimed to accelerate the execution time by focusing on the activation function's output that is strongly related to accuracy. In this study, we developed a technique to accelerate the execution time by stopping the learning model so that the activation function of the first and second layers would become zero. Two experiments were conducted in order to confirm the effectiveness of the proposed method. We first used the Iris dataset, implemented the proposed method, and compared its execution time with the execution time of Bayesian optimization. Based on the experimental results, the proposed method could accelerate the execution time by approximately 9s while maintaining the same precision. Therefore, we were successful in accelerating the execution time of Bayesian optimization for deep learning. Second, we applied our proposed method to analyze the credit card transaction data. Based on the experimental results, our method could accelerate the execution time by approximately 20s while maintaining the same precision. The results of the experiments demonstrate that the purpose of this study was achieved. In particular, we concluded that the proposed method can achieve a faster acceleration when deep learning is applied to an extremely large amount of data.

In the future, we will apply the proposed method to analyze the actual credit card transaction data and verify the acceleration effect with real large-scale data. We will also consider improving the Bayesian estimation algorithm.

ACKNOWLEDGEMENTS

The authors would like to thank Intelligent Wave Inc. for their comments on credit card transaction datasets.

This work was supported by JSPS KAKENHI Grant Number JP17K00310.

REFERENCES

- [1] Y. Bengio, "Learning deep architectures for AI," Foundations and Trends in Machine Learning, vol. 2, no. 1, pp. 1–127, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1561/2200000006
- [2] Q. Le, "Building high-level features using large scale unsupervised learning," in Acoustics, peech and Signal Processing (ICASSP), 2013 IEEE International Conference on, May 2013, pp. 8595–8598.
- [3] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout Networks," ArXiv e-prints, Feb. 2013.
- [4] S. Chib, "Markov chain monte carlo methods: computation and inference," Handbook of econometrics, vol. 5, pp. 3569–3649, 2001.
- [5] P. R. Conrad, Y. M. Marzouk, N. S. Pillai, and A. Smith, "Accelerating Asymptotically Exact MCMC for Computationally Intensive Models via Local Approximations," ArXiv e-prints, Feb. 2014.

- [6] Tensorflow. (Access Date: 6 April, 2017). [Online]. Available: https://www.tensorflow.org/
- [7] M. Lichman, "UCI machine learning repository," 2013, (Access Date: 6 April, 2017). [Online]. Available: http://archive.ics.uci.edu/ml
- [8] T. Minegishi and A. Niimi, "Proposal of credit card fraudulent use detection by online-type decision tree construction and verification of generality," International Journal for Information Security Research (IJISR), vol. 1, pp. 229–235, 2013.
- [9] A. Niimi, "Deep learning for credit card data analysis," in World Congress on Internet Security (WorldCIS-2015), Dublin, Ireland, Oct. 2015, pp. 73–77.
- [10] A. Niimi, "Deep learning with large scale dataset for credit card data analysis," in Fuzzy Systems and Data Mining II, Proceedings of FSDM 2016, ISO Press, Macau, China, Dec. 2016, pp. 149–158.

AUTHORS

Ayahiko Niimi is Associate Professor of Future University Hakodate, and Kousuke Sakamoto is master student of its University.



INTENTIONAL BLANK

THE FIVE LAYERS OF THE INTERNET ON THE COMPUTING LEVEL

Bing Li

Department of Computer Engineering, Xi'An Technological University, XiAn, China

ABSTRACT

To share huge amount of heterogeneous information, the Internet is reconstructed to consist of five layers, including routing, multicasting, persisting, presenting and humans. Routing layer establishes the fundamental substrate and locates resources with social disciplines. Multicasting layer disseminates data efficiently based on the routing. Persisting layer accesses persistent data with minimum dedicated resources. Presenting layer absorbs users' interactions to adjust the underlying layers through connected views to users. Different from the lower layers, the topmost one is made up with humans, which are social capital dominating the Internet. Within the upgraded infrastructure, besides the situation that a lower layer supports its immediate upper one, the humans layer influences the lower ones by transferring social resources to them. Those resources lead to adaptations and adjustments of the software layers since each of them needs to follow social rules. Eventually, the updated underlying layers return latest outcome to users upon those modifications.

KEYWORDS

Social Computing, World Wide Web, Peer-to-Peer computing, Software Architecture, Social Network Services, Information Sharing, Routing

1. INTRODUCTION

This document describes, and is written to conform to, author guidelines for the journals of AIRCC series. It is prepared in Microsoft Word as a .doc document. Although other means of preparation are acceptable, final, camera-ready versions must conform to this layout. Microsoft Word terminology is used where appropriate in this document. Although formatting instructions may often appear daunting, the simplest approach is to use this template and insert headings and text into it as appropriate.

It is believed that the Internet on the computing level should be reconstructed since it has never been designed in a proper way [1]. For its existing downsides, it is tough to share the huge amount of heterogeneous information to the large-scale of heterogeneous users. To upgrade, a five-layered infrastructure is proposed, which includes the routing, the multicasting, the persisting, the presenting and the humans from the bottom to the top, respectively.

Natarajan Meghanathan et al. (Eds) : ICAITA, Signal, SAI – 2017 pp. 15–25, 2017. © CS & IT-CSCP 2017

The term of the Internet on the computing level is defined as the Internet-based distributed environment concentrated on resource management and fundamental architectures to support high performance and low cost applications. As a counterpart beneath it, the Internet on the communication level aims to support raw data transmission between two nodes over the Internet. At present, TCP/IP is its primary representative of the level. This paper extends the definition of the computing level in the sense that all of the issues over TCP/IP are included in the scope. To simplify, when mentioning the term of the Internet in the paper, it specifies the one on the computing level.

The routing layer is responsible for managing and locating the tremendously huge resources over the Internet. Until now, a proper routing mechanism is still not available over the Internet. Although Google-like systems take this responsibility and win the huge successes in the industrial domain, they do not touch the essences. It is convinced that the routing over the Internet must follow the rules of social resources [2][3] rather than any other machine-based algorithms.

The multicasting layer is mandatory for the Internet system. To date, the protocol, HTTP (Hyper-Text Transmission Protocol), is still one of the most important solutions to transfer data over the Internet. It aims to transmit text-based or lightweight data. For that, it missed a lot of issues, such as the distributed topology, the transmission environment and the data characteristics, etc. When heavyweight [4] data or timing-sensitive [5] data are to be disseminated, the protocol becomes out of date. It is required to design a bunch of new algorithms based on the routing layer to distribute heavyweight and timing-sensitive data in the high performance and low cost manner.

The third layer is the persisting that supports high performance huge data saving and accessing over the Internet scale with minimum dedicated resources investments. To upgrade, it must be aware that the law of the socialized distributed environment is different from that of traditional machine-based ones. The utilization of undedicated resources [6] that are controlled by their proprietary owners is crucial to the persisting layer. In addition, it is also required to enhance the system with supports from the underlying layers, i.e., the routing and the multicasting.

The presenting layer demonstrates a fundamental application to users. The layer ameliorates the traditional WWW [7] in terms of raising the quality of data connections and supporting upgraded publishing and reading approaches. Moreover, the layer is also an approach to bind its upper layer with the underlying ones together. Traditionally, the WWW [7] plays the role, which aims to connect all of relevant data, i.e., Web pages. However, neither does it provide mandatory management approaches to build the linked world, nor does it guarantee the quality of the connections among pages such that it does not make sense to follow links to keep accessing referenced data.

The topmost layer is the one of humans, which does not exist in any traditional software infrastructures. In a conventional system, all of its subsystems run independently without human interventions, which is different from the dominant effects of the humans layer. The layer not only determines the outcome of the underlying layers but also governs their performance and cost. Its influences are transferred to each of the underlying layers although it owns only one direct link between itself and the routing layer, which is responsible for absorbing and managing social resources from the humans layer. Traditionally, a directed connection exists between two immediate neighboring layers, which play the roles of the source and the destination respectively from the lower to the upper. In contrast, the influences of the humans layer span all of the layers,

starting from itself, passing through the routing, the multicasting, the persisting, to the presenting finally.

It should emphasize that the paramount value is raised from humans on the Internet rather than any other traditional algorithms. For that, a new computing model is proposed. That is, while the resources of the systems include traditional ones, such as CPU cycles, storage and bandwidth, it is more important to introduce a new kernel to control those resources. The new kernel is the humans layer which provides the most precious value, i.e., the socialized capital [3][4], to the Internet.

The paper consists of the sections to present its primary ideas. Section 2 discusses the previous related researches in the domain and makes concise comparisons. Section 3 summarizes the generic requirements of users over the Internet. Section 4 explains in details about the layers of the Internet on the computing level. Section 5 articulates the collaboration of the layers. Section 6 gives a brief summary about the current and future work.

2. RELATED WORK

As Barabasi Albert-Laszlo [1] said researchers would not design the Internet like present if they knew its natures as deeply as nowadays.

2.1. The Routing

How to design a routing algorithm that helps each node to find colleagues and computing resources is critical for a large-scale information system over the Internet. Such a system is open, huge, heterogeneous and even unrevealed such that no dedicated coordinators are competent to handle the problem for the system.

2.1.1. Routing for Publishing

The goal of routing for publishing is to gain as many readers as possible. However, no any routing answers which users would be potential readers.

The problem is caused primarily by the fact that all of the users over the Internet are not registered uniquely. Moreover, it is tough to identify their interests even though their profiles are available. Some applications [8][9] over the Internet intend to solve the problem. They require users to have an account. Moreover, any node in the systems has followers or friends. Thus, when publishing, those users become the potential readers.

While the systems make progresses, the issue of routing for publishing still remains the old states in the overall infrastructure of the Internet. The exact routing algorithms should be refined by taking into account underlying social capital.

2.1.2. Routing for Reading

The questions on the issue of routing for reading are what is the appropriate way to present the huge amount of data to users and how to accomplish it, respectively.

When the Web was popular in 1990s, a directory service believes that data is exhibited in a hierarchical manner. However, it does not guarantee that the obtained pages meet users' requirements. Additionally, it is difficult to maintain a huge amount of heterogeneous Web sites manually. Even worse, the huge Web can hardly be abstracted to the static hierarchy of URLs upon a limited count of categories.

Then, the keyword searching, such as Google, takes the place of directory services. It aims to show data that matches users' keywords exactly or approximately. The service fulfills one of users' requirements of accessing information. However, it is far from a comprehensive solution. Different from traditional data retrieval approaches [10], it is expected that retrieved results form a particular structure upon their inherent senses rather than one sequence of irrelevant pages. Moreover, the searching consequences rely heavily on keywords from users. Over the Web, the quantity of pages is extremely larger than that of words of natural languages. Thus, Google-like systems work better only when searching by a sequence of keywords existing in pages rather than by a fuzzy term or a sequence of ones to be analyzed. Another problem is that the systems have to match all of indexed pages crawled from the Web and most matching is a waste of cost.

Some social network based systems [8][9] merge to overcome the difficulties. That is, data is presented in a context formed by a reader's friends or followees such that users keep highly active to read information from the context. The topic is tough since it associates with the difficult question, i.e., what on earth a large-scale of heterogeneous and homogenous data is interweaved [11][12]. If it was answered, the proper solution to presenting data could be proposed. The paper puts forward approaches to do that. Eventually, the solution is applied to the underlying infrastructure of the Internet on the computing level comprehensively in order to support the multicasting, the persisting and the presenting other than users.

2.2. The Multicasting

The multicasting is divided into two categories, the messaging and the streaming, according to the data transmission character. If data is transferred in a request/response manner, it belongs to the messaging protocol. Instead, if data is transmitted among nodes in the form of flowing from a source to a destination continuously, it is believed to be a streaming protocol.

2.2.1. The Messaging Protocol

The protocol of HTTP transfers data in the messaging way. Through it, the server usually encounters the difficulty to respond clients when receiving a large number of concurrent requests. Even worse, because of the richness of heavyweight data, the cost of the server becomes higher.

It is expected to upgrade the distributed environment and provide high quality routing services to accelerate the messaging.

2.2.2. The Streaming Protocol

To transmit heavyweight data, a bunch of competitors [4][5][13] are proposed. They attempt to form a decentralized topology to perform multicasting in a streaming way.

The problems of BitTorrent are found in its low quality routing to find collaborators. It is caused

by the fact that BitTorrent cannot detect the preferences of multicasting participants, especially the social relationships among them.

To multicast timing-sensitive heavyweight data, BiToS splits data into two sets, a high priority one and a low priority one [5]. Then, it performs BitTorrent algorithms on both of them concurrently. Therefore, although BitTorrent rarest-first algorithm is applied, the playback buffered data from the high-priority set is filled fast enough to become consecutive approximately. The problem of BiToS is the same as that of BitTorrent in terms of the lack of the high quality routing.

Tribler [13] can be regarded as Facebook or Twitter for multicasting. It finds potential collaborators from a buddy list before performing the multicasting with BitTorrent. However, the drawback of TRIBLER is caused by the buddy list as well. While it is better than the random routing of BitTorrent, it does not make sense that a user always has the same preference as its friends.

It should consider the approach to place users into a wider context other than circles of friends. Then, it provides a node with more opportunities to find potential collaborators based on either social relations or common interests. Moreover, it is believed that the routing for multicasting is an interactive and adaptive procedure on the fly other than a one-step task performed before data transmission.

2.3. The Persisting

The major problem to persist the huge amount of data over the Internet is to confirm the distinctive principles that determine the performance and the cost.

Cloud computing [14] answers the question by providing sufficient dedicated resources like data centers. It is traditional since it does not consider the natures of the Internet. It is a solution to enterprises rather than the Internet environment.

OceanStore [6] utilizes the undedicated resources from proprietary users like cafeteria owners by attracting them with business incentives. The approach lowers the cost to persist data with the participation of business-oriented resources. With the support of the non-technical design, the performance is raised to a high degree, especially for the popular data.

It is required to borrow the experiences of OceanStore in the persisting layer of the Internet on the computing level with in-depth understanding of the Internet. Social incentives and social capital other than business ones should be taken into in the design of the Internet infrastructure.

2.4. The Presenting

The World Wide Web [7] is the fundamental application of the Internet on the computing level. Without the support of an essential underlying infrastructure, the WWW becomes withered such that it has lost its value to present and manage the huge amount of data over the Internet.

Upon the World Wide Web, an author is authorized the privilege to connect one single term or one phrase inside Web pages to any other ones through the techniques of hyperlinks. Tim-Berners Lee [7] believed that relevant data was weaved together in this way such that readers can follow the links to access the Internet.

Unfortunately, it is seldom for an author to spend sufficient time adding links since it brings additional workload. Additionally, advertisements replace many high quality pages. A reader loses interests to navigate along the routes formed by the links because of that. As a matter of fact, WWW is being degenerated into a graph, which lacks significant connections.

2.5. Conclusions

In brief, the current Internet on the computing level was full of drawbacks in terms of its infrastructure. It results in high cost and low performance applications. Moreover, it is also a barrier to fulfill users' data accessing requirements. The Internet on the computing level should be redesigned upon the nature of the Internet to provide publishers and readers with an appropriate environment to access the large information world.

3. THE INFRASTRUCTURE OF THE INTERNET ON THE COMPUTING LEVEL

To resolve the problems of the current Internet on the computing level, it is indispensable to establish the New Infrastructure of the Internet on the Computing Level (NIICL), which reflects the natures of the Internet. The system consists of five layers, the routing, the multicasting, the persisting, the presenting and the humans, which are shown in Figure 1.

The term, the computing level, is used to differentiate from the one that supports fundamental communications between any two nodes over the Internet. Rather than the protocol of TCP/IP to exchange data physically, the new infrastructure contains all of the tasks based on sharing data logically.

3.1. The Differences

Different from classic layer-based software architectures that work only in the way of a lower layer supporting its immediately upper layer from the bottom to the top, NIICL is an interactive layered system. In the architecture, similar to traditional ones, a lower layer supports its immediately upper one. In addition, the routing layer is also visible to the humans one such that it has to adjust for crucial updates of the topmost layer. Within the fundamental infrastructure, all of the lower ones are under the domination of the topmost, the humans. The influences of the topmost one are transferred directly or indirectly to the underlying ones. For the distinctness of the system, it claims that the Internet should be involved with social resources other than computing resources. That is, such a system relies mainly on humans rather than computers. In another word, the social capital from the humans layer guarantees the quality of the Internet applications.

3.2. The Humans Layer

The layer of the humans is made up with all of the users of NIICL. According to their scales, the

users are divided into individuals and organizations. According to their roles, the users consist of publishers and readers. Users' behaviors online rely only their inherences, preferences, experiences, knowledge and common senses, i.e., human capital [2], which is accumulated during a long term.

The ways to capture the human capital is unavailable yet. For that, it is difficult to detect a user's tastes and intents with their actions directly. As a system to share huge information, it becomes tough to publish and read since it is difficult to find relevant targets, i.e., the readers and the data. Then, it is impractical for them to collaborate with one another and perform the Internet computing efficiently.

Fortunately, rather than human capital, social capital [2][3] or social resources are convenient to be captured. No matter whether to publish or read, users need to interact with others. It is feasible to retain the interactions by the concept of connections or neighbors. With the growth of the system, a huge graph is constructed.

In short, human capital from the humans layer dominates the Internet through users' interactions to access data. Although they are tough to be identified, it is convenient to map human capital to social capital through interactions among users.

3.3. The Routing Layer

The routing layer is defined as the mechanism to discern and locate sufficient resources for any other layers to support their activities in a high performance and low cost manner. To do that, it explores the routes between readers and authors. That is why the layer is named the routing. For a large-scale system like the Internet, it is expected that the routes be as short as possible to raise the performance.

One of the primary tasks of the layer is to discern social resources from the topmost, the humans layer. To do that, the concept of a connection is defined as the interaction between an author and a reader. As all of the users of the Internet perform publishing and reading following their preferences, it is time to retain those connections and construct a graph upon those interactions. It indicates that the topmost layer is associated with the bottom one directly, which results in the social graphs distinguished by their node types, data, circles and humans, respectively.

After the graphs are available, it is time to maintain, rank and find relevant resources upon them whenever publishing and reading are performed. The primary principle to do that is to identify those nearby nodes of a publisher or a reader since it is believed that they are the potential resources to fulfill the requirements of the publisher or the reader. That is, they are willing to accept data from the publisher or they hold data that the reader needs.

3.4. The Multicasting Layer

The multicasting layer is responsible for transmitting data to potential computing devices, which are located by the routing layer before the transmission is started as well as during the procedure of the transmission.

With respect to the characteristics of social networks [8][9], it is known that a large portion of data is published by an extremely small number of users, i.e., super nodes. Most users play the

role of readers as ordinary nodes. Therefore, it is a heavy load to send data from one publisher to its readers. The seeking is performed initially before the transmission is started. Once if those resources are available, an appropriate topology is constructed between a publisher and the located resources to balance the load and raise the performance.

In addition, it is critical to keep on routing on the fly since the potential resources are not stable in NIICL. All of resources participating in the transmission belong to proprietary users. Then, they join and leave the topology casually during the multicasting procedure. To lower the impacts of the departure, it is required to place those devices into a fault-tolerant graph rather than a tree. It is also necessary to keep seeking potential resources by interacting with the underlying routing layer during the entire procedure.

3.5. The Persisting Layer

The persisting layer aims to save huge amount of data into mass storage devices. The first problem is whether storage resources are dedicated or not. As a storage system for the Internet, it is impossible to provide sufficient dedicated resources because of the large scale. A reasonable system is made up with both dedicated and undedicated storage resources. It is critical to find undedicated resources from proprietary users, which are original publishers as well as forwarders. They would like to contribute their resources.

The next issue is where to persist. As a global system, the locations of data are critical to the performance. Since consumers of public data reside each corner of the globe, it is tough to answer the question. Fortunately, it is unnecessary to care about the issue because of the remarkable routing layer. Since readers might become potential resource providers such that their physical locations are identical to where the data should be persisted. For the nature of the routing layer, new users gain high performance from the previously persistent data. In the solution, physical locations are hidden by social positions [15], i.e., the social capital, to a great extent.

The final problem is how many replicas to persist. The number of data replicas is another critical factor to determine the performance. It is also difficult to estimate the number for the Internetbased storage system with traditional approaches. To resolve the problem, data is replicated and persisted to the owners who are willing to share with the assistance of the routing layer. During the procedure, the number of replicas is controlled by its original publisher and secondary publishers. The count of replicas depend on the popularity of data. In essence, the quality of data determines its quantity of replicas. Thus, it is unnecessary to take into account the issue either when persisting.

3.6. The Presenting Layer

The presenting layer is responsible for interacting with users and showing data to users. The layer aims not only to fulfill users' requirements but also to achieve the goal to cohere users with the system. Through the interactions, users become the dominators to guide the underlying layers to perform, adjust and optimize. It is the layer that converts human and social capital to govern the entire system. Its value is different from traditional presentations which separate users from computing systems and leave all of tasks to machines.

While the infrastructure is governed by the social graphs of data, circles and humans, they are updated on the fly following users' behaviors through the layer. Those behaviors consist of two

types, i.e., writing and reading. Both of the behaviors result in the connections for data, circles and humans. In brief, they are collected by the presenting layer and transferred to the underlying ones.

4. THE COLLABORATION OF THE LAYERS

The five layers form the new infrastructure of the Internet on the computing level (NIICL). They works in the traditional way the lower layer supporting its immediately upper one from the bottom to the top as well as they interact with one another controlled by social capital [15]. The collaboration among the layer is illustrated in Figure 1.



Figure 1. The Collaboration of the Five Layers

4.1. The Supporting From the Bottom to the Top

The routing layer supports its immediately upper one, the multicasting, through locating sufficient numbers of readers and publishers to publish data in a large scale. Moreover, a reader obtains accurate and comprehensive responses from relevant publishers and other readers.

The multicasting layer plays the crucial role to transmit data in a high quality way to support the persisting layer. Only if data is disseminated rapidly, the services of persisting data are performed efficiently in a large scale heterogeneous environment.

The persisting layer supports the presenting layer by keeping data in a proper manner in terms of nearby locations and rational replicas. That is the bedrock to present data in a high performance and low cost way. Although users and data are located in various physical places, both of them are presented in social locations.

The presenting layer shows connected data, circles and humans to users of the humans layer. It hides the underlying mechanism from users such that users gain a comfortable data accessing environment that matches users' personal preferences and social cognition.

4.2. The Interactions Among the Layers

24

The distinctive character of NIICL is that those layers interact with each other rather than support immediately upper layers only. It is caused by the core resources of the Internet on the computing level, i.e., human and social resources. The social graphs are always ameliorated by user behaviors in the humans layer, which is the dominant component of the infrastructure.

When users enter the system initially, they become members of the humans layer. Through the presenting layer, it is convenient to observe and locate the connected data, circles and humans they are interested in. Their behaviors are supported efficiently and effectively through those views. More important, it results in the updates of the connections surrounding them such that it impacts the consequences from the routing layer directly. Thereafter, the relevant multicasting and the persisting are improved for both of them depending on the routing. It is no doubt that it also results in updated connected views presented to users.

5. THE CURRENT AND FUTURE WORK

The research has been carried out for more than twelve years since November 2005. The research attempts to convert the existing WWW to the new infrastructure. During the procedure of the research, it is found that the critical problems over the Internet on the computing level are associated with diverse domains, such as social cognition, social network analysis, economics, statistical physics, library & information science, social computing and statistics, etc. The overall architecture is an important progresses to the current Internet.

As a huge system to share information, it is mandatory to place data in an appropriate form for publishers and readers. From my point of view, the traditional approaches do not fit in the complicated situation of the Internet. During the past years of the Internet development, the trend is apparent. That is, traditional Web sites, directory sites and even search engines are not as significant as social networks oriented systems even in the domain of information sharing. Information structures have innate ties with social capital. It is required to be clarified in the future research.

ACKNOWLEDGEMENTS

This research is sponsored by the Ministry of Education, Shaanxi Province, China. The number of the support is 14JK1358. I appreciate the support!

REFERENCES

- Albert-laszlo Barabasi. 2014. Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life. Publisher: Basic Books, ISBN-10: 0465085733, ISBN-13: 978-0465085736.
- [2] Yilei Shao. 2007. Exploring Social Networks in Computing Systems. PhD Dissertation, Princeton University, 2007.
- [3] James S. Coleman. 1988. Social Capital in the Creation of Human Capital. American Journal of Sociology, 1988, Volume 94 Supplement, Pages: 95-120.

- [4] Bram Cohen. 2003. Incentives Build Robustness In BitTorrent. 1st Workshop on the Economics of Peer-2-Peer System, 2003.
- [5] Aggeglos Vlavianos, Marios Iliofotou and Michalis Faloutsos. 2006. BiToS: Enhancing BitTorrent for Supporting Streaming Applications. Proceedings of 25th IEEE International Conference on Computer Communications, 2006, Pages: 1-6.
- [6] John Kubiatowicz, et al. 2000. OceanStore: An Architecture for Global-Scale Persistent Storage. ACM SIGPLAN Notices, Volume 35 Issue 11, Nov. 2000, Pages: 190-201.
- [7] W3C. 2000. World Wide Web Consortium. http://www.w3.org.
- [8] Matei Ripeanu. 2001. Peer-to-Peer Architecture Case Study: Gnutella Network. Proceedings of 1st International Conference on Peer-to-Peer Computing, 2001, Pages: 99-100.
- [9] Ian Clarke, Oskar Sandberg, Brandon Wiley and Theodore W. Hong. 2001. Freenet: A Distributed Anonymous Information Storage and Retrieval System. International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability, 2001.
- [10] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze. 2008. Introduction to Information Retrieval. Cambridge University Press, 2008, ISBN: 978-0-521-86571-5.
- [11] Mark Granovetter. 1983. The Strength of Weak Ties. American Journal of Sociology 78, 1983, Pages: 1360-1380.
- [12] Ronald S. Burt. 1992. Structural Holes: the Social Structure of Competition. Harvard University Press, 1992, ISBN: 0-674-84371-1.
- [13] J. A. Pouwelse, et al. 2008. Tribler: A Social-Based Peer-to-Peer System. Journal of Concurrency and Computation: Practice & Experience – Recent Advances in Peer-to-Peer Systems and Security (P2P 2006), Volume 20, Issue 2, February 2008, Pages: 127-138.
- [14] Qusay Hassan. 2011. Demystifying Cloud Computing. Journal of Defense Software Engineering, 2011 (January/February), Pages: 16-21.
- [15] Stephen P. Borgatti, Ajay Mehra, Daniel J. Brass and Gluseppe Labianca. 1994. Network Analysis in the Social Sciences. Science, 1994, 323(5916): 892-895.

AUTHORS

Bing Li is a faculty of Xi'An Technological University. I have been working here for four years since 2013. Before that, I worked as an associate professor for Peking University, the most famous university in China between 2005 and 2013. I got my PhD degree in the area of software engineering from Arizona State University in 2004. Moreover, I have some industrial working experiences in Lucent and IBM. He focused on the research in the area of the system-level software for Internet. He proposed the upgraded WWW and tried to program the entire system until now. Because of his rich experiences of programming on the



system-level of the Internet, he summarizes a series of APIs and design patterns for large scale distributed system programming, which is called the Great Free Cloud Programming Environment. His personal blog is located at http://greatfree.lofter.com.

INTENTIONAL BLANK

OPTIMAL SIZING AND PLACEMENT OF DISTRIBUTED GENERATORS AND CAPACITORS IN RADIAL DISTRIBUTION NETWORK

Partha P Biswas¹, P. N. Suganthan¹ and Gehan A J Amaratunga²

¹School of Electrical & Electronic Engg., Nanyang Technological University, Singapore
²Department of Engg., University of Cambridge, UK

ABSTRACT

Distribution network is an integral part of power system and it acts as the interface with consumer load points. The network incurs significant amount of losses in the electrical system. Loss minimization remains one of the prime objectives in distribution network optimization. This paper proposes an application of L-SHADE algorithm to simultaneously size and place both distributed generators (DGs) and shunt capacitors (SCs) in distribution network to reduce system real power loss. SHADE is the success history based parameter adaptation technique of differential evolution (DE). L-SHADE improves the performance of SHADE by linearly reducing the population size in successive generations. The algorithm is applied to minimize loss in standard IEEE 33-bus radial distribution network (RDN) and the simulation results are compared with some recent studies on the topic.

KEYWORDS

Radial distribution network, Distributed generator (DG), Shunt capacitor (SC), Power loss, L-SHADE algorithm

1. INTRODUCTION

Mitigating ever increasing load demand is one of the major challenges faced by utility companies. It may not always be feasible to boost capacity of transmission network. Locally installed distributed generators (DG) and added shunt capacitor (SC) banks in the system can augment the capacity, reduce losses, improve voltage profile and power quality of the network. Distributed generators can be a diesel generator, a wind turbine, solar photovoltaics (PV), fuel cells etc. Optimal sizing and siting of both distributed generators and shunt capacitors are of significance in improving system performance. Several literatures [1-3] focused on optimal sizing and placement of DGs only in pursuit of real power loss minimization. A network comprising both DG and SC has been studied in few literatures. Naik et al. [4] took analytical approach to optimally size and locate both the components. In most recent papers, heuristic methods such as hybrid harmony search algorithm (HSA) and particle artificial bee colony (PABC) [5], intersect mutation differential evolution (IMDE) [6] and back-tracking search algorithm (BSA) [7] have been applied for optimal design of capacity and placement of both distributed generators and shunt

Natarajan Meghanathan et al. (Eds) : ICAITA, Signal, SAI – 2017 pp. 27–37, 2017. © CS & IT-CSCP 2017

DOI: 10.5121/csit.2017.71303

capacitors. Ref. [8] optimizes size and location of both DG and SC in various networks with simultaneous minimization of both real and reactive power losses using decomposition based multi-objective evolutionary algorithm (MOEA/D).

The current study implements L-SHADE [9] algorithm to optimally locate and size both DG and SC in radial distribution network. SHADE [10] is a success history based parameter adaptation technique of DE optimization process for a constrained, multimodal non-linear problem. The convergence of the algorithm to global optima is fast and it potentially outperforms most other DE variants on CEC benchmark problems [9]. SHADE exhibited good performance in optimal power flow solutions [11]. In recent times L-SHADE has successfully been applied and has shown very competitive performance in windfarm layout optimization [12], in total harmonic distortion minimization of multilevel inverters [13], in hybrid active power filter parameter optimization [14] etc. Motivated by the growing application and noteworthy performance of L-SHADE in power domain, we apply the algorithm on the problem of radial distribution network (RDN). The distribution network is to be reinforced with optimally sized and appropriately placed DGs and SCs so that network real power loss is minimized. As an obvious fact, the problem is about simultaneous optimization of discrete variables i.e. locations (bus nos.) of all the components and continuous variables i.e. ratings of all the components. Further, the problem is non-linear due to the requirement of power flow calculation that involves numerous system components.

The organization of rest of the paper is done following way. Section 2 includes the mathematical formulation of power flow in distribution network. Section 3 describes the algorithm and its application. Section 4 discusses the case studies and simulation results. The paper ends with conclusion and possible future work in section 5.

2. MATHEMATICAL MODEL

The mathematical formula pertaining to power flow in the network are presented in this section with the aid of a simple feeder configuration.

2.1. Power flow formulation



Fig. 1. Single line diagram of a radial feeder [8]

Single line diagram of a simple feeder-line configuration is shown in Fig. 1. The computation of power flow is performed by following equations [8]:

$$P_{j+1} = P_j - P_{Lj+1} - R_{j,j+1} \cdot \frac{P_j^2 + Q_j^2}{|V_j|^2}$$
(1)

$$Q_{j+1} = Q_j - Q_{Lj+1} - X_{j,j+1} \cdot \frac{P_j^2 + Q_j^2}{|V_j|^2}$$
(2)

$$|V_{j+1}|^{2} = |V_{j}|^{2} - 2(R_{j,j+1}, P_{j} + X_{j,j+1}, Q_{j}) + (R_{j,j+1}^{2} + X_{j,j+1}^{2}) \cdot \frac{P_{j}^{2} + Q_{j}^{2}}{|V_{j}|^{2}}$$
(3)

where, real power and reactive power flowing out of bus j are P_j and Q_j respectively; P_{Lj+1} and Q_{Lj+1} are the real load and reactive load connected at bus j + 1. Line section between buses j and j + 1 has resistance $R_{j,j+1}$ and reactance $X_{j,j+1}$. $|V_j|$ is the voltage magnitude of bus j. The power loss in line segment connecting buses j and j + 1 is computed using following equation:

$$P_{Loss}(j, j+1) = R_{j,j+1} \cdot \frac{P_j^2 + Q_j^2}{|V_j|^2}$$
(4)

where, real power loss is defined by P_{Loss} . Total real power loss, the optimization objective in the network, is obtained by summing up all the line losses as follows:

$$TP_{Loss} = \sum_{j=0}^{m-1} P_{Loss}(j, j+1)$$
(5)

In this study, we consider DG units supplying real power with unity power factor e.g. photovoltaic systems, micro turbines etc. Therefore, if a DG, delivering power output of P_{DG} , is added to a bus (say *j*-th bus), the load in that bus changes from P_{Lj} to $(P_{Lj} - P_{DG})$. Similarly, if *k*-th bus in the system has inductive load of Q_k , an addition of Q_C unit of capacitor bank alters the reactive load to $(Q_k - Q_C)$. During the search process, the algorithm checks all possible locations with all probable ratings of the equipment to decide best combination that results in minimum power loss.

2.2. Constraints

Magnitude of any bus voltage must lie within specified limits of maximum and minimum voltages. Current in any branch shall not exceed the rated capacity of the branch. Mathematically, these can be written as:

$$V_{min} \le |V_j| \le V_{max} \tag{6}$$

$$\left|I_{j,j+1}\right| \le I_{j,j+1(\max)} \tag{7}$$

where, V_{max} and V_{min} are the maximum and minimum allowable voltages for any bus in the network. The numerical values of these parameters for the systems are considered as 0.90 p.u. and 1.05 p.u. respectively. $|I_{j,j+1}|$ is the magnitude of current flowing in the line linking bus *j* and bus j + 1, while $I_{j,j+1}(\max)$ is the maximum permissible current through the same branch considering the thermal capability limit of the line. It is worthwhile to mention that current carrying capacities of the branches in IEEE bus system are not explicit. Moreover, as installation of DGs and SCs improve the voltage profile of the network, the current reduces from the base configuration. Hence, verification of this constraint is not necessary.

29

3. L-SHADE ALGORITHM AND APPLICATION

Differential Evolution (DE) is a stochastic, population based optimization algorithm [15]. SHADE [10] is success history based adaptive DE where control parameters scaling factor (F) and crossover rate (CR) are automatically adjusted during the evolution process. Algorithm L-SHADE [9] is an extension of SHADE. In L-SHADE, the population size is dynamic and it reduces in successive generations following a linear function. The algorithm alongwith its applicationon RDN problem is briefly described in this section.

3.1. Initialization

Firstly, DE optimization process creates an initial population of probable solutions by assigning random values (within feasible bound) to each decision vector of the population. Initialization of j-th component of i-th decision vector is done as [16]:

$$x_{i,j}^{(0)} = x_{min,j} + rand_{ij}[0,1] * (x_{max,j} - x_{min,j})$$
(8)

where $rand_{ij}[0,1]$ is a random number lying between 0 and 1. The superscript '0' signifies initialization of population members.

3.2. Mutation

In next step during mutation process, donor/mutant vector $v_i^{(t)}$ is created corresponding to each population member or target vector $x_i^{(t)}$ in the current generation *t*. The mutation strategy used here is 'current-to-*p*best/1':

$$v_i^{(t)} = x_i^{(t)} + F_i^{(t)} \cdot \left(x_{pbest}^{(t)} - x_i^{(t)} \right) + F_i^{(t)} \cdot \left(x_{R_1^i}^{(t)} - x_{R_2^i}^{(t)} \right)$$
(9)

The mutually exclusive integers $R_1^i \& R_2^i$ are randomly chosen from the population range [1, Np]; $x_{pbest}^{(t)}$ is randomly selected from top 100p% ($p \in [0,1]$) individuals of current generation. The positive control parameter $F_i^{(t)}$ scales the difference vectors at *t*-th generation. The mutation strategy adopted in L-SHADE helps to exploit the search space efficiently and converge into an optimal solution.

3.3. Parameter adaptation

At each generation t, each individual has its own $F_i^{(t)}$ and $CR_i^{(t)}$ parameters that are used to generate the trial vector. Adaptation of these two parameters follows as:

$$F_i^{(t)} = randc(\mu F_r^{(t)}, 0.1)$$
(10)

$$CR_i^{(t)} = randn(\mu CR_r^{(t)}, 0.1)$$
⁽¹¹⁾

where $randn(\mu CR_r^{(t)}, 0.1)$ and $randc(\mu F_r^{(t)}, 0.1)$ are the sampled values from Normal and Cauchy distributions respectively. Normal distribution has a mean of $\mu CR_r^{(t)}$ and a variance of 0.1. Location and scale parameters of Cauchy distribution are $\mu F_r^{(t)}$ and 0.1 respectively. $\mu F_r^{(t)} \& \mu CR_r^{(t)}$ are randomly chosen from successful candidates of past generations saved in a

30

memory. The two values are initialized to 0.5 and subsequently modified by weighted Lehmer mean [9,10].

3.4. Crossover

Donor vector $v_i^{(t)}$ enters into the trial/offspring vector $u_i^{(t)} = (u_{i,1}^{(t)}, u_{i,2}^{(t)}, \dots, u_{i,d}^{(t)})$ by mixing its components with target vector $x_i^{(t)}$ through crossover. Binomial crossover is most commonly employed and it operates on each element based on adapted crossover rate $CR_i^{(t)}$. The scheme for an element is defined as:

$$u_{i,j}^{(t)} = \begin{cases} v_{i,j}^{(t)} \text{ if } j = j_{rand} \text{ orrand}_{i,j}[0,1] \le CR_i^{(t)}, \\ x_{i,j}^{(t)} \text{ otherwise} \end{cases}$$
(12)

where j_{rand} is a randomly chosen natural number in $\{1, 2, ..., d\}$, and d is the dimension of the decision vector.

3.5. Linear population size reduction

The success of SHADE algorithm is attributed to the adaptation technique of scale factor F and crossover rate *CR*. It has also been found that dynamic reduction in population size improves performance of SHADE. L-SHADE precisely implements the task by introducing a linear function for reduction of population size in successive generations. The population size starts with Np_{ini} (initial population size) and reduces closely matching the linear function before finally ending with Np_{min} (minimum population size). After each generation t, the population size in subsequent generation t + 1 is calculated by –

$$Np(t+1) = round\left[\left(\frac{Np_{min} - Np_{ini}}{NFE_{max}}\right) \cdot NFE + Np_{ini}\right]$$
(13)

 Np_{min} is set to 4 because mutation strategy adopted here requires minimum 4 individuals. NFE_{max} is the maximum number of fitness evaluations and NFE is the current number of fitness evaluations. If Np(t + 1) < Np(t), worst ranking individuals totalling [Np(t) - Np(t + 1)] are removed from the population [9]. A summary of steps involved in the optimization process is provided herein.

- A. Input and initialization:
 - 1. Input $Np_{ini} = 100$, $NFE_{max} = 20000$.
 - 2. Define vector x = [position, rating] and range of all these elements. We have 2 DG and 2 SC to size and allocate. So, a maximum of 4 elements for *position* (bus no.) and 4 elements for *rating* will form vector x (maximum, d = 8). Elements of position will always be rounded off to nearest integer.
 - 3. Create random initial population of 100 such vectors defined as x_i as per equation (8), i = 1, 2, ... 100.
 - 4. Set generation counter t = 0, dynamic population size $Np(t) = Np_{ini}$, evaluation counter NFE = 1 and control parameters $\mu F_r^{(0)} = \mu C R_r^{(0)} = 0.5$.
- B. Algorithm loop:

- 1. Evaluate $f(x_i^{(t)})$ i.e. ' TP_{Loss} ' in equation (5) for $x_i^{(t)}$ where i = 1 to Np. Increase counter *NFE* by Np i.e. NFE = NFE + Np.
- 2. while termination criteria $NFE < NFE_{max}$ do
- 3. **for***i* = 1 to *Np* **do**
- 4. Adapt control parameters $F_i^{(t)}$ and $CR_i^{(t)}$ as per equations (10) & (11).
- 5. Perform mutation to generate vector $v_i^{(t)}$ as per equation (9).
- 6. Perform crossover to generate element $u_{i,i}^{(t)}$ as per equation (12).
- 7. Evaluate $f(u_i^{(t)})$ i.e. ' TP_{Loss} ' for $u_i^{(t)}$. Increase evaluation counter NFE by 1 i.e. NFE = NFE+1.
- 8. Select best fit individuals for next generation. If, $f(u_i^{(t)}) \leq f(x_i^{(t)})$ and constraints in eq. (6) & (7) are satisfied, $x_i^{(t+1)} = u_i^{(t)}$. Else $x_i^{(t+1)} = x_i^{(t)}$.

End **for** loop.

- 9. Update population size for next generation Np(t + 1) as per LPSR strategy in equation (13).
- 10. Increase generation counter t = t+1. Go to step 2 of algorithm loop.

4. CASE STUDY RESULTS AND COMPARISON

IEEE 33-bus standard radial distribution network (RDN) diagram is provided in Fig. 2. Total load of the network is 3.72 MW and 2.3 MVAr. The maximum cumulative capacities of DGs (unity power factor) and capacitor banks in the installation are not to exceed 2 MW and 2 MVAr respectively. Table 1 summarizes the case description, results and comparison. 3 case studies are performed for the RDN. Case-1 deals with addition of only DGs in the network. Case-2 is study of the RDN when only SCs are added, while Case-3 considers both DGs and SCs. Average runtime in normal PC for most complex Case-3 is about 90 seconds for one complete run of the algorithm (i.e. $NFE_{max} = 20000$ function evaluations) on MATLAB platform. As can be seen from the tabulated results, increasing number of DG and SC reduces the system loss drastically. However, as in [6], we consider maximum 2 nos. of DG and 2 nos. of SCs that can be connected to the network. The loss data given by L-SHADE algorithm for all cases are the lowest. In Case-1, bus location and equipment ratings proposed by L-SHADE, IMDE and BSA are quite similar. However, bus-13 for allocation of one DG is preferred to bus-14 for effectiveness in loss reduction. In Case-2, cumulative ratings of SCs suggested by both IMDE and L-SHADE are almost equal. Again, bus-12 is advantageous location for one SC rather than bus-14. Ratings proposed by L-SHADE for SCs are higher in Case-3 when compared with IMDE algorithm. However, resulting network loss, the prime objective of optimization, is much reduced with little reshuffle in placement of both DG and SC.

32



Fig. 2: Standard IEEE-33 bus test system

Case	Parameter	Various optimization methods				
description		L-SHADE	IMDE [<u>6]</u>	Analytical [4]	Hybrid [5]	BSA [<u>7</u>]
Case-1	Realpower loss, kW	85.91	86.12 ¹	142.34	111.03	87.16
(DG only)	DG size, kW	845 (13),	840 (14),	1000(18)	2598(6)	851.6(13),
	(busno.)	1155 (30)	1130 (30)			1157.6(30)
	Min bus voltage in	0.9685(33)	0.9675 ¹ (33)	0.9311(33)	0.9425(18)	not reported
	p.u. (bus no.)					
Case-2	Realpowerloss, kW	135.75	139.7	164.6		-
(SC only)	SC size, kVAr	467 (12),	475 (14),	1000 (33)		-
	(busno.)	1039 (30)	1037 (30)			
	Min bus voltage in	0.9360(18)	0.942(18)	0.9165(18)		-
	p.u. (bus no.)					
Case-3	Realpower loss, kW	28.50	32.08	84.28	58.45	30.87
(DG + SC)	DG size, kW	829 (13),	1080 (10),	447 (18),	2531(6)	860 (13),
	(busno.)	1121 (30)	896.4 (31)	559(17)		1310.5 (30)
	SC size, kVAr	452 (12),	254.8 (16),	400 (33),	1250 (30)	334.8 (14),
	(busno.)	1041 (30)	932.3 (30)	500 (32)		899.8 (30)
	Min bus voltage in	0.9803 (25)	0.979 (25)	0.961 (30)	0.9536(18)	not reported
	p.u. (bus no.)					

Table 1. Simulation results for IEEE 33-bus radial distribution network

Fig. 3 shows convergence of L-SHADE for Case-3. As observed from the diagram, the algorithm converges in less than 8000 fitness evaluations. Linear population size reduction technique is demonstrated in Fig. 4 where the population size reduces almost linearly to 4 individuals. Fig. 5 indicates bus voltage profiles for various case studies performed in this literature. Voltages of all buses are within the allowable limits. Further, more uniform voltage profile is obtained when multiple components with smaller ratings are distributed throughout the network.

¹Values are recalculated with the proposed ratings of DG and SC [6]



Fig. 3: Convergence of L-SHADE algorithm for Case-3



Fig. 4: Linear population size reduction of L-SHADE algorithm for Case-3



Fig. 5: Bus voltage profile of 33-bus system for various case studies

5. CONCLUSION

The present paper discusses in detail the application and usefulness of L-SHADE optimization algorithm for deciding the rating and location of DG and SC simultaneously in the distribution network to reduce real (active) power loss. Size and location proposed by the algorithm for the equipment lead to lower system real power loss than the loss achieved by other equivalent algorithms. Reduction of loss by any amount is of technical and commercial advantage. Further the algorithm converges to the optimal solution very fast. The effectiveness of the algorithm in reducing loss in the networks with large number of buses remains the topic for future study.

ACKNOWLEDGEMENTS

This project is funded by the National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program.

REFERENCES

- [1] Viral, R., & Khatod, D. K. (2015). An analytical approach for sizing and siting of DGs in balanced radial distribution networks for loss minimization. International Journal of Electrical Power & Energy Systems, 67, 191-201.
- [2] Ayodele, T. R., Ogunjuyigbe, A. S. O., & Akinola, O. O. (2015). Optimal location, sizing, and appropriate technology selection of distributed generators for minimizing power loss using genetic algorithm. Journal of Renewable Energy, 2015.
- [3] Karimyan, P., Gharehpetian, G. B., Abedi, M., & Gavili, A. (2014). Long term scheduling for optimal allocation and sizing of DG unit considering load variations and DG type. International Journal of Electrical Power & Energy Systems, 54, 277-287.
- [4] Naik, S. G., Khatod, D. K., & Sharma, M. P. (2013). Optimal allocation of combined DG and capacitor for real power loss minimization in distribution networks. International Journal of Electrical Power & Energy Systems, 53, 967-973.

- [5] Muthukumar, K., & Jayalalitha, S. (2016). Optimal placement and sizing of distributed generators and shunt capacitors for power loss minimization in radial distribution networks using hybrid heuristic search optimization technique. International Journal of Electrical Power & Energy Systems, 78, 299-319.
- [6] Khodabakhshian, A., & Andishgar, M. H. (2016). Simultaneous placement and sizing of DGs and shunt capacitors in distribution systems by using IMDE algorithm. International Journal of Electrical Power & Energy Systems, 82, 599-607.
- [7] Fadel, W., Kilic, U., & Taskin, S. (2016). Placement of Dg, Cb, and Tcsc in radial distribution system for power loss minimization using back-tracking search algorithm. Electrical Engineering, 1-12.
- [8] Biswas, P. P., Mallipeddi, R., Suganthan, P. N., & Amaratunga, G. A. (2017). A multiobjective approach for optimal placement and sizing of distributed generators and capacitors in distribution network. Applied Soft Computing.
- [9] Tanabe, R., & Fukunaga, A. S. (2014, July). Improving the search performance of SHADE using linear population size reduction. In Evolutionary Computation (CEC), 2014 IEEE Congress on (pp. 1658-1665). IEEE.
- [10] Tanabe, R., & Fukunaga, A. (2013, June). Success-history based parameter adaptation for differential evolution. In Evolutionary Computation (CEC), 2013 IEEE Congress on (pp. 71-78). IEEE.
- [11] Biswas, P. P., Suganthan, P. N., & Amaratunga, G. A. (2017). Optimal power flow solutions incorporating stochastic wind and solar power. Energy Conversion and Management, 148, 1194-1207.
- [12] Biswas, P. P., Suganthan, P. N., & Amaratunga, G. A. (2017, June). Optimal placement of wind turbines in a windfarm using L-SHADE algorithm. In Evolutionary Computation (CEC), 2017 IEEE Congress on (pp. 83-88). IEEE.
- [13] Biswas, P. P., Awad, N. H., Suganthan, P. N., Ali, M. Z., & Amaratunga, G. A. (2017, June). Minimizing THD of multilevel inverters with optimal values of DC voltages and switching angles using LSHADE-EpSin algorithm. In Evolutionary Computation (CEC), 2017 IEEE Congress on (pp. 77-82). IEEE.
- [14] Biswas, P. P., Suganthan, P. N., & Amaratunga, G. A. (2017). Minimizing Harmonic Distortion in Power System with Optimal Design of Hybrid Active Power Filter using Differential Evolution. Applied Soft Computing.
- [15] Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE transactions on Evolutionary Computation, 13(2), 398-417.
- [16] Biswas, P. P., Suganthan, P. N., & Amaratunga, G. A. (2017). Optimization of Wind Turbine Rotor Diameters and Hub Heights in a Windfarm Using Differential Evolution Algorithm. In Proceedings of Sixth International Conference on Soft Computing for Problem Solving (pp. 131-141). Springer, Singapore.

36

AUTHORS

Partha P. Biswas

Partha P. Biswas received his Bachelor of Electrical Engineering degree from Jadavpur University, Kolkata, India in the year 2002. He was awarded MSc in Power Engineering from Nanyang Technological University, Singapore in 2016. He has worked in different engineering companies at various positions for about 13 years in Singapore and India. Currently he is pursuing his Ph.D. in the School of Electrical and Electronic Engineering, Nanyang Technological University. His research topic is optimization of power system using evolutionary algorithms.

Ponnuthurai N. Suganthan

Ponnuthurai N. Suganthan received the B.A. degree, the Postgraduate Certificate, and the M.A. degree in electrical and informationengineering from the University of Cambridge,U.K., in 1990, 1992, and 1994 respectively. He received the Ph.D. degree inelectrical and electronic engineering from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore in 1996. Heis currently an Associate Professor in the School of Electrical and Electronic Engineering, Nanyang Technological University include evolutionary computation, applications of evolutionary computation, neural networks, pattern recognition and bioinformatics.

Gehan A. J. Amaratunga

Gehan A. J. Amaratunga received the B.S. degree in electrical engineering from Cardiff University, Wales, U.K., and the Ph.D. degree in electrical engineering from University of Cambridge, U.K. He has been holding the 1966 Professorship in Engineering at the University of Cambridge since 1998, where he is currently the Head of the Electronics, Power, and Energy Conversion Group, Electrical Engineering Division, Department of Engineering. He has worked for 20 years on integrated and discrete electronic devices for power conversion and on the science and technology of carbon-based electronics for 15 years. He also leads an active research effort in novel solar-cell technologies. Dr. Amaratunga is a Fellow of the Royal Academy of Engineering, U.K



37

INTENTIONAL BLANK

MULTI-VIEW FEATURE FUSION NETWORK FOR VEHICLE RE-IDENTIFICATION

Haoran Wu, Dong Li, Yucan Zhou, and Qinghua Hu

School of Computer Science and Technology, Tianjin University, Tianjin, China

ABSTRACT

Identifying whether two vehicles in different images are same or not is called vehicle reidentification. In cities, there are lots of cameras, but cameras cannot cover all the areas. If we can re-identify a car disappearing from one camera and appearing in another in two adjacent regions, we can easily track the vehicle and use the information help with traffic management. In this paper, we propose a two-branch deep learning model. This model extracts two kinds of features for each vehicle. The first one is license plate feature and the other is the global feature of the vehicle. Then the two kinds of features are fused together with a weight learned by the network. After, the Euclidean distance is used to calculate the distance between features of different inputs. Finally, we can re-identify vehicles according to their distance. We conduct some experiments to validate the effectiveness of the proposed model.

KEYWORDS

vehicle re-identification, deep learning model, feature fusion

1. INTRODUCTION

With the development of economy, urban traffic management becomes extremely important. Usually, large cities have many monitoring cameras, and these cameras can be used for vehicles tracking. However, for economic and privacy reasons, it is difficult for these monitoring cameras to cover all areas. In this situation, re-identifying a vehicle when it disappears from one camera and appears in another is necessary for tracking. In addition to that, vehicle re-identification also benefits to criminal investigation, urban construction and so on.

Usually, there are two problems in re-identification: finding a feature used for re-identification and finding a suitable method for distance metrics. The feature should be robust in different situations. The distance metrics method aims to learn a distance which is able to distinguish inner class and inter class. In vehicle re-identification based on monitor video, we can use license plates to identify whether two cars are the same or not, as the license plate is the only identification of a vehicle. Certainly, there has been a lot of effective license plate recognition methods proposed in recent years. Some methods use traditional image processing techniques [1, 2] and some of them take use of the neural network [3]. However, in real situation, the license plates are not always identified. When identifying license plate, some plates are too fuzzy to be recognized as shown in Fig. 1 (a). There are also some vehicles without license plates like Fig. 1 (b). In condition of

Natarajan Meghanathan et al. (Eds) : ICAITA, Signal, SAI – 2017 pp. 39–47, 2017. © CS & IT-CSCP 2017

DOI: 10.5121/csit.2017.71304

traffic congestion, some vehicles are shielded by vehicles in front of them like Fig. 1 (c). In these conditions, license plates are useless for vehicles re-identification. Thus, a more common feature is required for vehicle re-identification. Since Geoffrey and Alex put forward the AlexNet in 2012, it is popular to use deep neural network for extracting features [4, 5, 6]. For the vehicle images in which the license plates are hard to be recognized, we can use a deep neural network to extract global vehicle features. However, different from person re-identification, the vehicles of the same type usually have little difference. In this situation, the deep neural features maybe useless for vehicle re-identification.



Fig. 1. Sample images. (a) is a car whose plate is fuzzy. (b) is a vehicle without plate. (c) is a car whose plate is shielded by the car in front of it.

In order to solve the problems above, we propose a deep neural network, called Multi-View Feature Fusion Network (MVFFN). MVFFN is a two-branch network. This net aims to adapt to the following situations: license plates are hard to be recognized and vehicle types are similar. It fuses license plate features and vehicle features together according to different situations. If the effect of license plate features is small, the weights of license plates features will be small and the weights of vehicle features will be heavier. On the contrary, when re-identifying two similar cars, the weights of vehicle features will be small and weights of license plate features will be big. Experiments are conducted on dataset collected by ourselves and the result proves our method to be very effective.

The rest of the paper is organized as follows: Section 2 will give a review of related work. Section 3 will introduce the model and the specifically designed fusion method. Section 4 describes the experiments and results. The conclusion will be drawn in Section 5.

2. RELATED WORK

2.1. Feature Fusion Network

The previous re-identification works focus on person re-identification, and the purpose of person re-identification is similar to vehicle re-identification. Therefore, the researches of person re-identification are helpful to our work.

In the past, the person re-identification techniques usually rely on some handcrafted features, like color and texture histogram [10, 11]. With the development of deep neural network, there are also some work using deep neural networks [8, 12]. In the work, the Feature Fusion Network (FFN) [8] combines handcrafted features and deep neural features together, which gives us inspiration.

FFN extracts features by a deep neural network and hand-crafted method ELF [9], respectively. Then the two kinds of features are fused in the fusion layer. The output of the fusion layer is calculated by:

$$Z(x) = h(W^T x + b) \tag{1}$$

where x is the concatenation of two kinds of features and h is the ReLu function. W and b are weights and bias, respectively.

Then the output is sent to a softmax loss layer. Finally, the features extracted from the fusion layer are used to calculate the distance between different inputs. The distance metric method is Mirror Kernel Marginal Fisher Analysis (Mirror KMFA) [9].

2.2. Deep Relative Distance Learning

There are some previous researches on vehicles re-identification. The PROVID [17] also uses the vehicle features and license plate features, but the vehicle features are extracted by conventional methods. Besides, the PROVID is a layered structure. The most related work of vehicle re-identification we found is Deep Relative Distance Learning [13]. It proposes a Deep Relative Distance Learning (DRDL) model. The idea of this method is: two vehicles are different if the models of them are different, if they are vehicles of the same model then other features are applied. This model is based on VGG_CNN_M_1024. It has two branches, the first branch is VGG followed by an attribute recognition loss. This branch is designed to identify whether the inputs are vehicles of the same model or not. The other branch is VGG and coupled cluster loss specially designed for this problem. This branch is used to distinguish vehicles belong to the same model. Two kinds of features are concatenated and sent to another coupled cluster loss.

The inputs of this network are two image sets: a positive set and a negative set. The coupled cluster loss are specially designed for the relative distance learning problem. The loss function is

$$L(W, X^{P}, X^{N}) = \sum_{i}^{N^{P}} \frac{1}{2} \max\{0, \left\|f(x_{i}^{P}) - c^{P}\right\|_{2}^{2} + \alpha - \left\|f(x_{*}^{n}) - c^{P}\right\|_{2}^{2}\}$$
(2)

where c^{P} is the center of the positive set, which is the mean value of all positive samples. And x_{*}^{n} is the nearest negative sample to the positive set center point. α is a predefined constant parameter. When the distances of all the positive samples to the center point adds α are less than the distance of the nearest negative sample to the center point, the loss will be zero. On the other hand, the loss will be

$$\sum_{i=1}^{P} \frac{1}{2} \left(\left\| f(x_{i}^{P} - c^{P}) \right\|_{2}^{2} + \alpha - \left\| f(x_{*}^{n} - c^{P}) \right\|_{2}^{2} \right).$$
(3)

Inspired by their work, we also designed a two branch network to extract different features and fuse them together to get a more robust feature for vehicle re-identification.

3. METHODOLOGY

3.1. Network structure

The structure of the network is shown in Fig. 2. We formulate the problem as a binary classification problem just as the work by E. Ahmed et al. [14]. Therefore, we designed a network with a pair of inputs. The label of the paired inputs is 1 or 0 which represents whether they are the same or not. Each input in the inputs pair contains two images: one is the vehicle image and the other is the license plate image extracted from vehicle image by Faster R-CNN [7]. These four images are sent to VGG16 and get four 4096-dimension vectors. Then the features of the two license plates are sent to FC layer in which they are used to calculate the weight. After that, the vehicle feature and license plate feature are sent to the concatenate layer with the weight. In this layer, vehicle features and license plate features are connected together with the weight. Output of

concatenate layer is an 8192-dimension vector. Finally, each branch will get an 8192-dimension feature fusion vector, and the two feature fusion vectors are used to calculate Euclidean distance. The Euclidean distance is used to judge whether the pair of inputs are from the same vehicle or not.



Fig. 2. Multi-View Feature Fusion Network for vehicles re-identification

3.2. Feature Fusion

The features of the input images are extracted by VGG16, the outputs of VGG16 are 4096dimension vectors. The vehicle features and license plate features of the first input are $\{a_1, b_1\}$. The vehicle features and license plate features of the second input are $\{a_2, b_2\}$. For each input, we use a weight to integrate the vehicle feature and the license plate feature. Therefore, we design a FC layer to calculate the weight.

The input of this layer is

$$x = b_1 - b_2, \tag{4}$$

x is different between the two license plate features. The output of this layer is

$$w = f(W^T x) \tag{5}$$

where W is a 4096-dimension vector, f is the sigmoid function, and w is the weight calculated by this layer.

After we obtain the weight W, the vehicle feature and the license plate feature are fused in the concatenation layer. They are fused as

$$feature = [(1-w)*a, w*b] \tag{6}$$

The output of the concatenation layer is an 8192-dimension vector.

3.3. Contrastive loss

The loss function of this model is the contrastive loss. The contrastive loss is

$$L = \frac{1}{2N} \sum_{n=1}^{N} yd^{2} + (1-y) \max(m \arg in - d, 0)^{2}$$
(7)

where y is the label of the input pair and margin is the threshold. The distance d between two fused features is calculated by

$$d = \left\| feature_1 - feature_2 \right\|_2^2 \tag{8}$$

where $feature_1$ and $feature_2$ represent fused features of two inputs, respectively.

4. EXPERIMENT

4.1. Data sets

There are few open datasets in this area. The only dataset we found is VehicleID [13], there is not license plate information, it is unsuitable in our experiment. We collect a new dataset for vehicle re-identification. The dataset is collected from real-world video, which is obtained from a busy street in Tianjin, China. Some images of this dataset are shown in Fig. 3.

In this dataset, there are several images of one vehicle, so it is suitable for the vehicle reidentification task. The split of this dataset is given in Table 1.



Fig. 3. Images in dataset. Each line is the images of one vehicle.

Table 1. Data split for our dataset

Class number	Train images	Probe images	Gallery images
	number	number	number
389	4276	1222	2334

43

4.2. Training strategies

44

The training pairs of images are chosen from the training images. The positive pairs and negative pairs ratio is 1 to 3. Meanwhile, 10 percent of the training data are used for validation.

We use mini-batch stochastic gradient descent (SGD) [15] in our experiment. The initial learning rate is set as $\gamma = 0.001$, and the learning rate decreases by $\gamma_{new} = \gamma / number _epoach$. The initialized method of weights of FC layer is random initialization.

4.3. Weights distribution in MVFFN

The weights used to fuse two kinds of features are variable with the change of input pairs of images. That is, according to different situation, the network can learn a suitable weight to fuse the two kinds of features.

Some weights are shown in Table 2. and the corresponding gallery license plate images are shown in Fig. 4.

The first probe license plate is the plate of the first car, the 3rd gallery license plate belongs to the same car and it is clear, so the weight of the 3rd gallery is very high. The 13th gallery license plate is relatively obscure, in this condition, the weight is small. The 91st gallery license plate belongs to an unlicensed vehicles, so the weight will also be small. The 162nd gallery license plate is clear, so the weight is large.



Table 2. Some weights of the first license plate images

Fig. 4. Weights and corresponding gallery license plate images

4.4. Experiment results and analysis

To verify the effectiveness of our method, we design three experiments. We use the vehicle features and license plate features to re-identify vehicles, respectively. Another method is to concatenate the vehicle features and license plate features together. We use the cumulative match curve (CMC) [16] to evaluate the experimental results. The CMC of different methods is shown in Table 3. And the match rates from top-1 to top-10 are illustrated in Fig. 5.

From the results we can see that it is effective to use license plate information to re-identify vehicles. The vehicle information is not as powerful as license plate information but it is also useful especially when the license plate information is missing. If we just concatenate two features together, the top-1 matching rate of concatenation method is less than that of using license plate features by about 5%. The result shows that feature fusion can lead to a worse result without a good algorithm. Our method produces the best performance in each level. It beats the second-rank method by 1.48 percent in top-1 matching rate. We can get from the experiment results that our method is effective in the real environment.

Method	Top1	Тор3	Top5	Top10
Vehicle	58.43%	68.09%	72.67%	79.13%
License plate	92.55%	95.99%	97.22%	98.45%
Vehicle + plate	86.17%	91.82%	94.03%	95.99%
Ours	94.03%	97.22%	98.61%	98.94%

Table 3. CMC of Vehicle Re-identification task



Fig. 5. CMC of vehicle re-identification

Vehicle features are the most basic features for vehicle re-identification. However, in the real world, vehicle features between similar vehicles are hard to be distinguished. Besides, the background, viewpoint and illuminations will also influence the results. License plate features are suitable for this work and it gets a high accuracy rate. But for these reasons mentioned in Section 1, license plate features are useless in certain scenarios. Feature fusion is a good idea, but a proper fusion algorithm is needed. With our method, we can get a weight with each input pair of images.

When the license plate information is missing, the vehicle information will be more important. On the contrary, if the license plate information can re-identify vehicles with high confidence the weight of vehicle information will be smaller.

5. CONCLUSION

In this paper, a Multi-View Feature Fusion Network is proposed for vehicle re-identification. This model aims to solve two problems in the real environment: vehicles of the same model is difficult to re-identify and in some cases the license plate information is missing. This model combines two kinds of features (vehicle features and license plate features) with a weight w. The weight can be trained according to different situations. Compared with other features, the fused features achieve higher predict accuracy. Experimental results show that the fusion strategy is effective and useful.

REFERENCES

- Nijhuis, J. A. G., Ter Brugge, M. H., Helmholt, K. A., Pluim, J. P. W., Spaanenburg, L., & Venema, R. S., et al. (1995). Car license plate recognition with neural networks and fuzzy logic. IEEE International Conference on Neural Networks, 1995. Proceedings (Vol.21, pp.2232-2236 vol.5). IEEE Xplore.
- [2] Naito, T., Tsukada, T., Yamada, K., & Kozuka, K. (2000). Robust license-plate recognition method for passing vehicles under outside environment. Vehicular Technology IEEE Transactions on, 49(6), 2309-2319.
- [3] Anagnostopoulos, C. N. E., Anagnostopoulos, I. E., Loumos, V., & Kayafas, E. (2006). A license plate-recognition algorithm for intelligent transportation system applications. IEEE Transactions on Intelligent Transportation Systems, 7(3), 377-392.
- [4] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. International Conference on Neural Information Processing Systems (Vol.25, pp.1097-1105). Curran Associates Inc.
- [5] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., & Anguelov, D., et al. (2014). Going deeper with convolutions. 1-9.
- [6] Arora, S., Bhaskara, A., Ge, R., & Ma, T. (2013). Provable bounds for learning some deep representations. Eprint Arxiv, 584-592.
- [7] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis & Machine Intelligence, 39(6), 1137-1149.
- [8] Wu, S., Chen, Y. C., Li, X., Wu, A. C., You, J. J., & Zheng, W. S. (2016). An enhanced deep feature representation for person re-identification. 1-8.
- [9] Lai, J., Lai, J., & Lai, J. (2015). Mirror representation for modeling view-specific transform in person re-identification. International Conference on Artificial Intelligence (pp.3402-3408). AAAI Press.
- [10] Farenzena, M., Bazzani, L., Perina, A., Murino, V., & Cristani, M. (2010). Person re-identification by symmetry-driven accumulation of local features. Computer Vision and Pattern Recognition (Vol.23, pp.2360-2367). IEEE.
- [11] Li, Z., Chang, S., Liang, F., Huang, T. S., Cao, L., & Smith, J. R. (2013). Learning Locally-Adaptive Decision Functions for Person Verification. IEEE Conference on Computer Vision and Pattern Recognition (Vol.9, pp.3610-3617). IEEE Computer Society.

46

- [12] Yi, D., Lei, Z., & Li, S. Z. (2014). Deep metric learning for practical person re-identification. Computer Science, 34-39.
- [13] Liu, H., Tian, Y., Wang, Y., Pang, L., & Huang, T. (2016). Deep Relative Distance Learning: Tell the Difference between Similar Vehicles. Computer Vision and Pattern Recognition (pp.2167-2175). IEEE.
- [14] Ahmed, E., Jones, M., & Marks, T. K. (2015). An improved deep learning architecture for person reidentification. Computer Vision and Pattern Recognition (pp.3908-3916). IEEE.
- [15] Bottou, L. (2012). Stochastic Gradient Descent Tricks. Neural Networks: Tricks of the Trade. Springer Berlin Heidelberg.
- [16] Gray, D., Brennan, S., & Tao, H. (2007). Evaluating appearance models for recognition, reacquisition, and tracking.
- [17] Liu, X., Liu, W., Mei, T., & Ma, H. (2016). A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance. European Conference on Computer Vision (pp.869-884). Springer International Publishing.

AUTHORS

Haoran Wu is a student of Tianjin University, and this work was done when Haoran Wu was a postgraduate student.



INTENTIONAL BLANK

ACUTE LEUKEMIA CLASSIFICATION USING CONVOLUTION NEURAL NETWORK IN CLINICAL DECISION SUPPORT SYSTEM

Thanh.TTP¹, Giao N. Pham¹, Jin-Hyeok Park¹, Kwang-Seok Moon², Suk-Hwan Lee³, and Ki-Ryong Kwon¹

¹Dept. of IT Convergence and Applications Eng., ²Dept. Electronics Eng.,Pukyong National University, Busan, Korea ³Dept. of Information Security, Tongmyong University, Busan, Korea

ABSTRACT

Leukemia induced death has been listed in the top ten most dangerous mortality basis for human being. Some of the reason is due to slow decision-making process which caused suitable medical treatment cannot be applied on time. Therefore, good clinical decision support for acute leukemia type classification has become a necessity. In this paper, the author proposed a novel approach to perform acute leukemia type classification using convolution neural network (CNN) classifier. Our experimental result only covers the first classification process which shows an excellent performance in differentiating normal and abnormal cells. Further development is needed to prove the effectiveness of second neural network classifier.

KEYWORDS

Acute Leukemia Classification, Convolution neural network, Clinical decision support system

1. INTRODUCTION

Leukemia is a group of cancers that begin in the bone marrow and result in high numbers of abnormal white blood cells. It attacks white blood cells which responsible for immune system and disease prevention in the human body. There are four main types of Leukemia which are classified based on severity level and infected cells type - acute lymphoblastic leukemia (ALL), acute myeloid leukemia (AML), chronic lymphocytic leukemia (CLL) and chronic myeloid leukemia (CML). In this paper, we would like to propose a novel approach to perform acute leukemia classification based on Convolution Neural Network (CNN). This method provides an excellent performance in classification process that reaches 96.43% of accuracy to discriminate normal and abnormal cell images from given database.

2. RELATED WORK

2.1. OVERVIEW OF LEUKEMIA

A. Acute Lymphocytic Leukemia (ALL)

Acute Lymphocytic Leukemia (ALL) is one between two classes of acute leukemia which develop from early (immature) forms of lymphocytes cells in bone marrow. ALL is recorded as the most common of the 4 major types among children but the least common types in adults, but most deaths occur in adults which are understandable because children's bodies can often handle aggressive treatment better than adult's. Most leukemia infected white blood cells has average Natarajan Meghanathan et al. (Eds) : ICAITA, Signal, SAI – 2017

Natarajan Megnanathan et al. (Eds) : ICATTA, Signal, SAT = 2017pp. 49= 53, 2017. © CS & IT-CSCP 2017

DOI: 10.5121/csit.2017.71305

size about two times the red blood cells that surround them, while normal white blood cells have an average size that similar with the surrounding red blood cells. In most of ALL infected cells, the nucleus area occupies almost 80-90% of the whole cell and left about 20-30% of cell area for the cytoplasm. The surface of ALL cell is smooth.

B. Acute Myeloid Leukemia (AML)

Acute Myeloid Leukemia is developed from early (immature) forms of myeloid cells in bone marrow. AML cases generally occur in older people and uncommon before the age of 45. AML cases hold the most number of deaths among the 4 types of leukemia. The nucleus area is about 50-60% of the whole cell area and about 40-50% cell area is cytoplasm. Sometimes short purple stripes which called "Auer rods" can be found in the cytoplasm area and it makes AML identification become easier.

C. Chronic Lymphocytic Leukemia (CLL)

Chronic Lymphocytic Leukemia is a type of leukemia that attacks the lymphocytes, different from ALL which attack the immature form of lymphocytes. CLL cell has a similar characteristic with ALL with a small area of cytoplasm and almost 80-90% of the cell area is occupied by nucleus area. The difference between ALL and CLL is the existence of nucleoli inside the nucleus area. Because CLL is basically mature cells which development has been completed, there should not be any nucleolus detected inside CLL cells.

D. Chronic Myeloid Leukemia (CML)

Chronic Myeloid Leukemia is another type of chronic leukemia that attacks the myeloid cells and has a similar behavior like CLL, especially in the development state. The average age at diagnosis of CML is around 64 years old. CML cell has a similar characteristic with AML in terms of cytoplasm area ratio, CML also show a big portion of cytoplasm in the cell and most of CML cells are fully developed into mature white blood cells which make it as the most differentiable cells among the four leukemia types. CML nucleus has developed into differentiable shape and texture unlike AML which in most cases the shape is round and under developed.



Figure 1. (a) Sample images of Acute Lymphocytic Leukemia, (b) Acute Myeloid Leukemia, (c) Chronic Lymphocytic Leukemia, and (d) Chronic Myeloid Leukemia.

2.2 RECENT LEUKEMIA DETECTION AND CLASSIFICATION METHODS

The researches about leukemia classification in recent years based on computer vision. The most common algorithm in this approach consists of several rigid steps: image pre-processing, clustering, morphological filtering, segmentation, feature selection or extraction, classification, and evaluation [1] [2]. Almost authors have used machine learning techniques to detect blood cell

in images and to classify cells in images. They extract features representing points, regions, or objects of interest and then use those features to train a model to classify or learn patterns in the image data. Feature extraction usually involves processing each image with some image-processing operations, such as calculating gradient to extract the discriminative information from each image. In this paper, we used the method of deep learning to learn characteristics of leukemia shape to classify normal and abnormal cell images.

3. PROPOSED METHOD

In this paper, we will use a Convolution Neural Network (CNN) to perform classification and extract features from raw images.

3.1. ARCHITECTURE OF CNN

In this work, we proposed a network contains 4 layers. The first 3 layers for detecting features and the other two layers (Fully connected and Softmax) are for classifying the features. The input image has the size [50x50x3]. The receptive field (or the filter size) is 5x5. The stride is 1 then we move the filters one pixel at a time. The zero-padding is 2. It will allow us to control the spatial size of the output image (we will use it to exactly preserve the spatial size of the input volume so the input and output width and height are the same). During the experiment, we found that in our case, altering the size of original image during the convolution lead to decrease the accuracy about 40%. Thus the output image after convolution layer 1 has the same size with the input image.

The convolution layer 2 has the same structure with the convolution layer 1. The filter size is 5x5, the stride is 1 and the zero-padding is 2. The number of feature maps (the channel or the depth) in our case is 30. If the number of feature maps is lower or higher than 30, the accuracy will decrease 50%. By experiment, we found the accuracy also decrease 50% if we remove Convolution layer 2.

The Max-Pooling layer 25x25 has Filter size is 2 and stride is 2. The fully connected layer has 2 neural. Finally, we use the Softmax layer for the classification.



Figure 2. The architecture of network.

4. EXPERIMENT AND RESULT

In this experiment, the original ALL-IDB1 image database which consists of 108 cell image [3] (59 normal cell images and 49 abnormal cell images) was used. The number of labelled training data and unlabelled testing data are shown in Table 1.

The experiment conducts on Matlab and the accuracy rate of the recognition of leukemia by our proposed CNN model achieved 96.43%.

	Training Set	Test Set
Normal cell	40	19
Abnormal cell	40	9
Total	80	28

Table 1. The number of training and test data.

5. CONCLUSION

Leukemia is one of the most fatal diseased for the human being. This paper proposed a novel acute leukemia type classification method using CNN as one of the constructing modules for the acute leukemia type clinical decision support system. In the future study, we will consider how to improve our architecture to obtain a better result and try to classify four types of Leukemia.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research F oundation of Korea (NRF) funded by the Ministry of Science and ICT (No. 2016R1D1A3B03931003, No. 2017R1A2B2012456), MSIP (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2017-2016-0-00318) supervised by the IITP (Institute for Information & Communications Technology Promotion)", Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2015-0-00225), the Pukyong National University Research Park(PK-URP) for Industry-Academic Convergence R&D support program 2017, which is funded by the Busan Metropolitan City, Korea, and the Brain Busan 21 Project in 2017.

REFERENCES

- I. Vincent, L. Yang, J.H. Park, P.N. Giao, S.H. Lee, and K.R. Kwon, "Classifying Normal and Abnormal White Blood Cells for Early Detection of Leukemic Disease using Curvelet Transform," in Proceeding of Workshop on Image Processing and Image Understanding 2014, 2014.
- [2] L.Putzu and C.D.Ruberto, "White Blood Cells Identification and Counting from Microscopic Blood Image," World Academy of Science, Engineering and Technology, vol. 73, 2013.
- [3] R.D.Labati, V.Piuri, and F.Scotti, "ALL-IDB: The Acute Lymphoblastic Leukemia Image Database for Image Processing," in 18th IEEE International Conference on Image Processing, 2011.
- [4] Vasuki Shankar, Mohan Deshpande, N Chaitra, and S Aditi, "Automatic detection of Acute Lymphoblastic Leukemia using Image Processing," in International Conference on Advances in Computer Applications (ICACA), 2016.
- [5] Chun-Mei Huo, Hong Zheng, Hong-Yi Su, Zhao-Liang Sun, Yi-Jin Cai, and Yi-Fei Xu, "Tongue Shape Classification Integrating Image Processing and Convolution Neural Network," in 2017 2nd Asia-Pacific Conference on Intelligent Robot Systems, 2017.
- [6] Shrutika Mahazan, Snehal S. Golait, Ashwini Meshram, and Nilima Jichlkan, "Review: Detection of types of Acute Leukemia," International Journal of Computer Science and Mobile Computing, vol. 3, no. 3, pp. 104-111, 3 2014.

52

AUTHORS

Tran Thi Phuong Thanh received the B.S. degree in Information Technology from Water Resources University, Vietnam, in 2015. She is currently studying a master course in IT Convergence and Application Engineering in Pukyong National University, Korea. His research interests include digital signal and image processing, bioinformatics.

Jin-Hyeok Park received the B.S. degree in Computer Software Engineering from Dongeui University, Korea in 2013 and he received an M.S. degree in IT Convergence and Application Engineering in Pukyong National University, Korea in 2015. He is currently studying Ph.D. course in IT Convergence and Application Engineering in Pukyong National University, Korea. His research interests include digital signal and image processing and data compression.

Kwang-Seok Moon He received the B.S., or Studies M.S., and Ph.D degrees in Electronics Engineering in Kyungpook National University, Korea in 1979, 1981, and 1989 respectively. He is 1currently a professor in department of Electronic engineering at Pukyong National University. His research interests include digital image processing, video watermarking, and multimedia communication.

Suk-Hwan Lee received a B.S., an M.S., and a Ph. D. Degrees in Electrical Engineering from Kyungpook National University, Korea in 1999, 2001, and 2004 respectively. He is currently an associate professor in Department of Information Security at Tongmyong University. His research interests include multimedia security, digital image processing, and computer graphics.

Ki-Ryong Kwon received the B.S., M.S., and Ph. D. degrees in electronics engineering from Kyungpook National University in 1986, 1990, and 1994 respectively. He worked at Hyundai Motor Company from 1986-1988 and at the Pusan University of Foreign Language from 1996-2006. He is currently a professor in Dept. of IT Convergence & Application Engineering at the Pukyong National University. He has researched University of Minnesota in the USA on 2000-2002 with Post-Doc, and Colorado State

University on 2011-2012 with visiting professor. He was the General President of Korea Multimedia Society on 2015-2016, also was a director of IEEE R10 Changwon section on 2012-2016. His research interests are in the area of digital image processing, multimedia security and watermarking, bioinformatics, weather radar information processing.









AUTHOR INDEX

Ayahiko Niimi 01 Bing Li 15 Dong Li 39 Gehan A J Amaratunga 27 Giao N. Pham 49 Haoran Wu 39 Jin-Hyeok Park 49 Ki-Ryong Kwon 49 Kousuke Sakamoto 01 Kwang-Seok Moon 49 Partha P Biswas 27 Qinghua Hu 39 Suganthan P. N 27 Suk-Hwan Lee 49 Thanh.TTP 49 Yucan Zhou 39