Experiment and Implementation of a Machine-Learning Based k-Value Prediction Scheme in a k-Anonymity Algorithm

Kumbayoni Lalu Muh† · Sung-Bong Jang††

ABSTRACT

The k-anonymity scheme has been widely used to protect private information when Big Data are distributed to a third party for research purposes. When the scheme is applied, an optimal k value determination is one of difficult problems to be resolved because many factors should be considered. Currently, the determination has been done almost manually by human experts with their intuition. This leads to degrade performance of the anonymization, and it takes much time and cost for them to do a task. To overcome this problem, a simple idea has been proposed that is based on machine learning. This paper describes implementations and experiments to realize the proposed idea. In thi work, a deep neural network (DNN) is implemented using tensorflow libraries, and it is trained and tested using input dataset. The experiment results show that a trend of training errors follows a typical pattern in DNN, but for validation errors, our model represents a different pattern from one shown in typical training process. The advantage of the proposed approach is that it can reduce time and cost for experts to determine k value because it can be done semi-automatically.

Keywords : Artificial Neural Networks, k-Anonymity, k Value Prediction, TensorFlow

k-익명화 알고리즘에서 기계학습 기반의 k값 예측 기법 실험 및 구현

Kumbayoni Lalu Muh† · 장 성 봉††

요 약

빅 데이터를 연구 목적으로 제3자에게 배포할 때 프라이버시 정보를 보호하기 위해서 k-익명화 기법이 널리 사용되어 왔다. k-익명화 기법을 적용할 때, 해결 해야할 어려운 문제 중의 하나는 최적의 k값을 결정하는 것이다. 현재는 대부분 전문가의 직관에 근거하여 수동으로 결정되고 있다. 이러한 방식은 익명화의 성능을 떨어뜨리고 시간과 비용을 많이 낭비하게 만든다. 이러한 문제점을 해결하기 위해서 기계학습 기반의 k값 결정방식을 제안한다. 본 논문에서는 제안된 아이디어를 실제로 적용한 구현 및 실험 내용에 대해서 서술 한다. 실험에서는 심층 신경망을 구현하여 훈련하고 테스트를 수행 하였다. 실험결과 훈련 에러는 전형적인 신경망에서 보여지는 패턴을 나타냈으며, 테스트 실험에서는 훈련에러에서 나타나는 패턴과는 다른 패턴을 보여주고 있다. 제안된 방식의 장점은 k값 결정시 시간과 비용을 줄일 수 있다는 장점이 있다.

키워드 : 인공 신경망, k-익명화, k값 예측, 텐서플로우

1. Introduction

Big Data analysis has become increasingly important to predict the future and make decisions about work in organizations. To enable such analysis, Big Data must be securely published to an analyzer. A typical case that requires securing data publishing is medical data. Medical data include private information, which if accessed by a hacker can be dangerous. Traditionally, the easiest way to protect critical private information would be to delete it from the original data before publishing.

However, when this approach is used, there are two types of problems. The first problem is that the distributed data can be rendered useless because some important information could have been removed unintentionally. In most cases, the recipient wants to uncover meaningful knowledge and information by analyzing the data. For example, a receiving researcher wants to calculate statistics of people in their twenties for a disease like cold. However, this might not be possible if the distributer deleted the disease

※ 이 연구는 금오공과대학교 학술연구비로 지원되었음(2017-104-025).
† 비 회 원 : 금오공과대학교 IT융복합공학과 석사과정
†† 정 회 원 : 금오공과대학교 산학협력단 부교수
Manuscript Received : October 11, 2019
First Revision : November 26, 2019
Second Revision : December 31, 2019
Accepted : December 31, 2019
* Corresponding Author : Sung-Bong Jang(sungbong.jang@kumoh.ac.kr)

attributes because they wanted to protect the information. As a result, the disease's statistical results could not be acquired. The second problem is the linkage attack by a hacker. In this attack, the hacker can acquire private critical information by combining separate data. For example, consider a case where the hacker already bought a database that included a person's address, name, gender, and age. Then, the hacker obtained medical data that had been anonymized by deleting the name, so it only included gender, age, address, and disease. However, if the hacker combined the two separate sources of information, they can derive the critical information and determine that "John Smith has cancer".

To solve these problems, k-anonymity and l-diversity have been devised and applied to various areas of data. In k-anonymity, the linkage attack probability is reduced by shielding the original information. For example, the birth date could be changed to a range of several years (see the 'birth' columns in Table 1) and the zip code could be anonymized by using only the initial digit (see the 'zip' columns in Table 1). By doing that, different data records would now appear to be the same, and the probability of identification would be reduced.

Table 1. An Example of k-anonymity with k=2

ID	birth	gender	zip	disease
1	1985	male	53703	cold
2	1990	female	53706	headache
3	1988	male	54078	stomache
4	1995	male	52582	cold

(a) Before Anonymization

ID	birth	gender	zip	disease
1	1985〉=	male	5*****	cold
2	1990〉=	female	5****	headache
3	1988〉=	male	5****	stomache
4	1990〉=	male	5****	cold

(b) After Anonymization with *k*=2

However, there are some disadvantages. One challenge is how broad to set the range. If the range is too high, the data utility becomes lower even though it increases the protection. On the other hand, if the range is too low, protection is lowered while data utility becomes higher. Finding an optimum value is a difficult task. Typically, this has involved an expert who has had much experience with the data. However, if the expert retires from a working company or moves to another company, there is no one who would be able to support this practice. This would lead to degrading the performance of the anonymization, and sometimes could even lead to huge information loss.

Until now, there has been almost no research to address this. To overcome the problem, a simple idea that is based on artificial neural network (ANN) has been proposed in our previous research [01]. This paper extends the idea raised in that paper and describes an experiment and implementation to realize the proposed idea.

In this work, much contributions have been done to previous works. Main contributions are as follows. First, in this work, the whole processes are presented for estimating k values based on machine learning. In our previous works, only simple idea has been proposed. Second, artificial neural networks model has been defined and it is trained and tested. In this model, dataset of three features are used to estimate k values. Third, the model has been implemented using Tensorflow libraries. In this work, several models are defined and evaluated by measuring RMSE errors. Then, optimal model has been chosen to predict optimal k value.

This work is organized as follows: Section 2 describes the previous research about k-anonymization; Section 3 presents a new method that is based on an ANN, describes an implementation, and reviews an experiment to evaluate the proposed solution; finally, Section 4 presents the conclusion.

2. Related Works

A summary of research about k-anonymization was covered in our previous paper. The basis of k-anonymization technique comes from the work of L. Sweeny [01] where the concept of k-anonymity was proposed. The basic idea is to change the original data into other values while minimizing information loss. Therefore, it doesn't use data deletion; instead, it uses methods such as generalization, where the original data is adapted using a hierarchical generalization tree that is structured using information about the data.

After this invention, many other approaches have been derived from this idea. B.C.M Fung et al. [03][04] presented enhanced k-anonymization schemes that were based on the time data. In their approach, they used timing information to enhance the performance of the k-anonymization. Also, they pointed out that there are many difficulties when applying the k-anonymization scheme to original source data. One issue is that it takes a great deal of time to anonymize the original data, and thus, the preparation may take an unreasonable amount of time (hundreds of years in extreme cases). To solve this problem they presented an approach based on adaptive k-anonymization. Nivedita Elanshekhar et al. [05] presented a method that is based on slicing adapted from the suppression technique. The purpose of the method was to enhance the privacy of Big Data while guaranteeing data utility. The approach consisted of six steps: Tuple Partitioning, Optional Column Generalization (Map), Attribute Partitioning (Reduce), and Suppression Slicing. Mohamed Nassar [06] proposed a Query language for data anonymization. Since data anonymization is a cumbersome task for database administrators because there are many things to be considered, they developed a tool similar to a database Query language. The tool was composed of three components. The first component was a request handler in order to divide a user's input query into two separate queries. The second component was a database adapter that was in charge of mediating a database management system (DBMS) and a request handler. The DBMS's are different from company to company; hence, a mediator is indispensable to adapt a DBMS. The last component was a data anonymizer which was responsible for anonymizing the data transferred from the request handler. It implemented and contained algorithms such as k-anonymity and l-diversity. They have implemented a prototype using Java and Hadoop server, and they conducted an experiment to evaluate their approach. Roberta Matsunaga [07] discussed how to ensure data privacy in Big Data analytics based on cloud computing. They insisted that the existing data anonymization techniques could be used efficiently for that purpose. Furthermore, they proposed an ontology definition for data anonymization. The ontology is one of the dictionaries that offers a common understanding of data concepts in order to provide users with guidelines and standards for efficient data anonymization. Xuyun Zhang [08] pointed out a local recording problem when k-anonymization applied to Big Data. To solve the problem, they presented a proximity privacy model. The solution considered semantic similarity among sensitive attribute values during big data anonymization. The advantage of the scheme was that it was easy to make the model more scalable. Also, they implemented the algorithms with MapReduce for parallel execution.

Research related to Artificial Neural Networks has been underway since the fifties. The basic idea is rooted in machine learning. In machine learning, the goal is to make a machine have a learning capability like a human being. The concept are described in works of [9-11]. In that work, the authors presented a basic scheme for machine learning with artificial intelligence. The scheme asserts that a machine can learn something through long-time training .

Machine learning is an important background technology that accelerates the advancement of artificial intelligence. There are two types of machine learning: supervised learning and unsupervised learning [12, 13]. In supervised learning, a neural network is trained by using target answers, and in unsupervised learning, a network is trained using the data with no answers. In our work, a neural network will be trained based on supervised learning. By this training, the weighted values in a neural network would be changed and optimized as illustrated in Fig. 1.

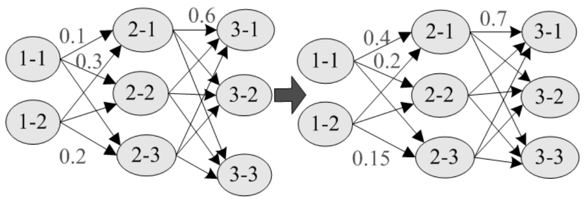

Fig. 1. An Example of Weights Updates in a Artificial Neural Network

In the figure, the circle represents a node and the arrow or edge represents a connection between nodes. The value on an edge shows a assigned value. In the example, a ANN consists of two number of input layer, three number of hidden layer, and three number of output layer. In the example, a weight value is used to calculate the next weight value. A weight value of a current node is computed by a weighted averaged summation. The formula for node 2-1 is as follows.

$$WAS(Node_{2-1}) = \sum_{k=1}^{2} [Value(Node_{1-k})* \\ Weight(Node_{1-k}, Node_{2-1})] + Bias \quad (1)$$

Here, weight values are changed whenever training repeats. The right hand-side figure represents a changed weight values after neural network training.

Machine learning can be achieved by training the Artificial Neural Network (ANN) [14, 15]. To enable the training of a neural network, high-speed computer processing is required because it takes a great deal of effort in order to analyze the data. This has been accomplished recently in [16]. Around 20 years ago computational time for training was thought of as an NP-hard problem [17]. However, thanks to a high-speed CPU and parallel processing technologies, training can be realized now within a short time.

Big Data has also contributed to the evolution of Artificial Intelligence (AI). In the past, AI correctness was substantially low because there was insufficient data to train a neural network. However, during recent decades a tremendous amount of data has been accumulated. By using this data a neural network can be trained enough to achieve correctness approaching that of a human being. Gustavo Carneiro et al. [18] has investigated the efficiency of machine learning in terms of input number and output number of nodes for a medical imaging analysis. He presented a method based on many variables for assessing the efficiency of a learning machine.

3. Proposed Approach

This section describes a proposed modified process. In the existing anonymizing process, the value of k is heuristically determined by system administrators, but the decision by the administrator may have been made without enough consideration of the performance and information loss. This can lead to degradation of the anonymization quality. To solve this problem, a new method is presented. A comparison between the proposed and a conventional method is illustrated in Fig. 2.

Fig. 2-(a) represents the existing process for Big Data anonymization, and Fig. 2-(b) shows the proposed scheme. The proposed process consists of six steps. The proposed modified steps for big data anonymization are as follows. First, the original data is read into the system, and a data administrator analyzes source data using a tool before transmitting the data to an anonymizer. If there is missing or invalid data, it must be fixed before proceeding. In most cases, the input data format might be a proprietary relational database (RDB) that is inherent to the DBMS provider.

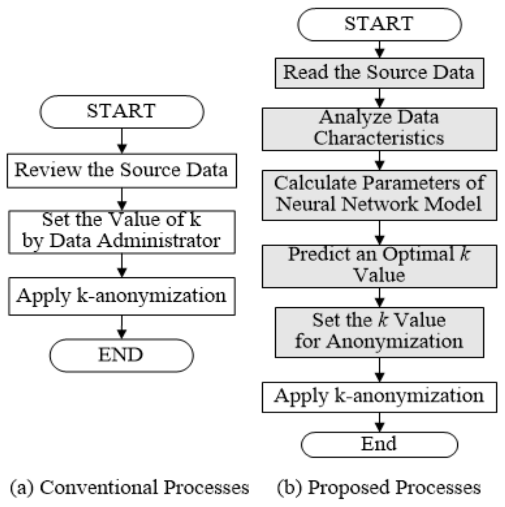
START
Review the Source Data
Set the Value of k by Data Administrator
Apply k-anonymization
END
START
Read the Source Data
Analyze Data Characteristics
Calculate Parameters of Neural Network Model
Predict an Optimal k Value
Set the k Value for Anonymization
Apply k-anonymization
End
(a) Conventional Processes
(b) Proposed Processes

Fig. 2. A Comparison of Conventional Scheme and Proposed Scheme

Second, it is necessary to analyze the basic characteristics of the source data such as the purpose, domain, and criticality. (Each of these three characteristics will be discussed in the following section in detail.) In the third step, the basic parameters for k value prediction are calculated using analyzed characteristics. In fourth step, a k value is predicted using a trained artificial neural network model. (The neural network model for prediction will be explained later in more detail.) After this, the value of k is set, and then Big Data anonymization is performed.

An example of a neural network model is illustrated in Fig. 3. The model has one input layer, two hidden layers, and one output layer. The Input layer has three input nodes that each receive input values. In the model, three features have been used for input data that includes allowed loss ratio level, private information protection level, and data criticality level.

The descriptions for each feature are as follows. The first feature represents an allowed level of loss ratio. This represents an allowed level of data loss when k-anonymity scheme is applied to the original data set to be anonymized. In k-anonymity scheme, source data are changed into generalized data before distribution to protect private information from linkage attack. During this process, some data value are lost. In general, there are trade-off relationships between k-anonymization and information loss. Many researches have been done to alleviate the information loss. One of reseaches is a scheme that

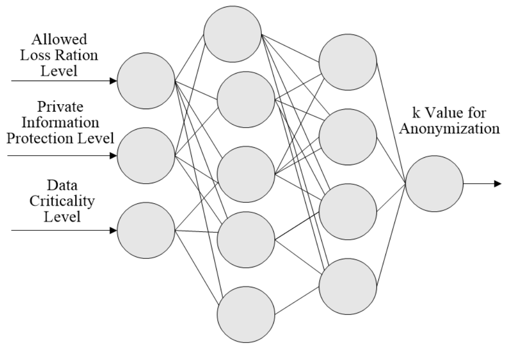
Allowed Loss Ration Level
Private Information Protection Level
Data Criticality Level
k Value for Anonymization

Fig. 3. Architecture of Neural Networks Model Defined in a Proposed Approach

is based on loss ratio control. In the scheme, they use a threshold value of loss. Depending on this value, they control the value k. Likewise, we define the permitted level of loss ratio as illustrated in Table 2.

Table 2. Permitted Level of Loss Ratio

Information Loss Ratio	Assigned Value
Very Low	1
Low	2
Middle	3
High	4
Very High	5

The second feature represents a private information protection level. This feature represents an allowed level of how much information should be protected before applying k-anonymiztion. In k-anonymization, if the value of k is set to be high, private information protection becomes high. But, information loss becomes high because more data should be converted and distorted. Hence, to consider appropriate protection level, this feature was added and collected to train a neural network model. In the same way of the allowed loss ratio level, assigned value ranges from 1 to 5.

The third feature is a data criticality level. This feature represents how much critical information are included in the original source data to be anonymized. The examples of critical information are identification number, disease information, and biological information. If the original data contains such information, then, high value is assigned to criticality level. The concept is based on the observation that if there are no critical information in the source data, we don't need to anonymize the data because there are no information to protect. For example, let's compare two tables as shown in Table 3.

Table 3. An Example of Published Data

Seq	Hobby	Favorite Food	Height	Weight
1	Football	Rice	155 cm	67
2	Baseball	Beef	178 cm	78
3	Swimming	Noodle	167 cm	56
.	.	.	.	.

(a) Patient Table in a Central Hospital

ID	birth	gender	zip	disease
1	1985	male	53703	cold
2	1990	female	53706	headache
3	1988	male	54078	stomache
4	1995	male	52582	cold

(b) Patient Table in a University Hospital

The first table contains less critical information than those in a second table because there is no disease information. Therefore, we can say the first is less critical, and thus give lower priority to the first. In the same way of the other two features, assigned value ranges from 1 to 5.

Using these three features, a data set must be collected to train a neural network model. However, there are no such data because no one tried to use neural network for prediction of k value. So, we had to make artificial data based on the heuristics. The data are shown in Table 4.

Table 4. Sample Data Set

Information Loss Ratio	Data Protection Level	Data Criticality	k value
3	2.3	4	3
1.5	3.5	2.7	2
3.2	4.6	2.4	4
.	.	.	.
.	.	.	.

Using this data, we have trained several neural network models and have chosen an optimal model from them for prediction. The whole process is shown in Fig. 4.

The first step is to define the structure of the Artificial Neural Network model. In the definition the number of neurons in the input layer, the hidden layer, and the output layer are determined, and it is in this stage where the values of the initial parameters such as default weight, learning rate, and activation function are determined. In the next step, the system reads training data that includes previous information

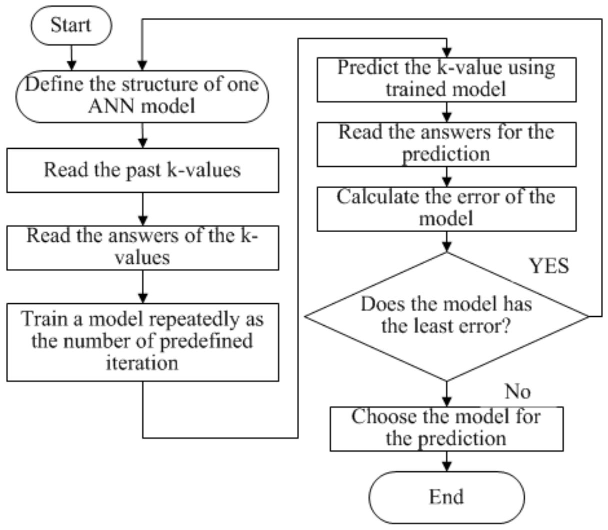
Start
Define the structure of one ANN model
Read the past k-values
Read the answers of the k-values
Train a model repeatedly as the number of predefined iteration
Predict the k-value using trained model
Read the answers for the prediction
Calculate the error of the model
Does the model has the least error?
YES
No
Choose the model for the prediction
End

Fig. 4. Determination Process of an Optimal Neural Network Model

loss ratios, data protection levels, and data criticality. In step three, the system takes the answer that includes the value of k. In step 4, the model is trained repeatedly. Through this step, the weight values of each edge are optimized according to the back-propagation algorithm. By doing this, the prediction capability of the model can be improved dramatically. In step 5, using the trained model the k values are predicted as well as the number of the output neurons. In step 6, we collect the real answers for the predicted period. In step 7, we calculate the error metric of the model using predicted values and real answer values. Then we compare this model with the previously tested models to determine if this latest model has the least error. The most widely used metric to check the error is Root Mean Square Error (RMSE). It can be calculated using an equation (2).

$$Error\ Of\ Model = \sqrt{\frac{1}{m}\sum_{k=1}^{m}(P_k - A_k)^2} \qquad (2)$$

In the equation, Pk represents a predicted value, and Ak is the real answer. The final step is to determine the model by analyzing the experimental results come from the previous step.

To support all these processes, we have implemented ANN algorithms using the TensorFlow Machine Learning platform as shown in Fig. 5.

TensorFlow is one of the most widely used tools for neural network analysis. First, we define a neural network model that has the input nodes, hidden

```
url = 'input_data.csv'
names= ['LOSS_RATIO','PROTECTION_LEVEL',
                  'CRITICALITY_LEVEL','TARGET']
df = pd.read_csv(url,names=names)
tf.logging.set_verbosity(tf.logging.ERROR)
y = df[['TARGET']]
x = df[['LOSS_RATIO','PROTECTION_LEVEL',
'CRITICALITY_LEVEL']]
k_prediction_model = Sequential()
k_prediction_model.add(Dense(48,input_dim=3,act='relu'))
k_prediction_model.add(Dense(48,activation='relu'))
k_prediction_model.add(Dense(1))
k_prediction_model.compile(loss='mean_squared_error',
                              optimizer='adam')
tensorboard=TensorBoard(log_dir='./logs',histogram_freq=0,write
_graph=True,write_images=False)
history = k_prediction_model.fit(x_train, y_train,
    validation_data=(x_temp,y_temp),batch_size=5,
    epochs=1000,callbacks=[tensorboard])
pred = k_prediction_model.predict(x_temp)
score = np.sqrt(metrics.mean_squared_error(pred, y_temp))
print("RMSE: {}" .format(score))
```

Fig. 5. An Example of Tenforflow Codes Implemented in a Experiment

nodes, output nodes, and related parameters. The TensorFlow code for data reading, a model definition, and training are shown in Fig. 5.

In this example, a single layer of a neural network is defined. It has the three nodes in the input layer, forty-eight nodes in the hidden layer, and one number in an output layer (As shown in Model 6 in Table 3). The iteration number for training in this example has been set to be 1000. In this step, many candidate models were defined. After training and evaluation, one model was selected as the final target model with the least error. The example of model definitions used for k prediction is illustrated in Table 5.

Table 5. Candidate Neural Network Model for k determination

Neural Network Model for Prediction of k value	Number of Nodes in the input layer	Number of Nodes in Hidden Layer	Number of Nodes in Output Layer
Model 1	3	6	1
Model 2	3	9	1
Model 3	3	12	1
Model 4	3	24	1
Model 5	3	36	1
Model 6	3	48	1

For each model, we calculate the RMSE errors and processing time. Table 6 shows the experimental results.

As shown in the results, RMSE errors and processing time have trade-off relationships. However, the processing time is not overly large even in the case of model 6. So, we have chosen model 6 as the final neural network model. The results of training error measurements are shown in Fig. 6.

Table 6. RMSE Errors and Processing Time for Each Model

Neural Network Model for Prediction of k value	Error of Model (RMSE)	Processing Time (Seconds)
Model 1	49.3456	4.34554
Model 2	47.9374	4.76788
Model 3	42.4237	5.43532
Model 4	35.2846	7.38478
Model 5	26.9741	13.2314
Model 6	23.0842	19.0943

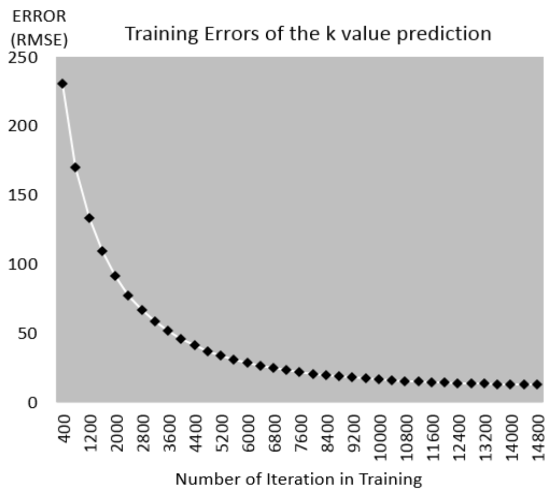

Fig. 6. Training Errors Distribution in an Experiment

In Fig. 6, the x-axis represents the number of training iterations and the y-axis shows the training error measurement. As shown in the figure, the trend of the training errors follows a well-known neural network error function that exponentially decreases. When the number of iterations reaches about 10,000, the error value becomes stable at 19.435.

Fig. 7 illustrates the measurement results of the validation errors. As can be seen from the figure, the results show a different pattern. In the results of the training errors shown in Fig.6, each error value simply decreases whenever the iteration number increases.

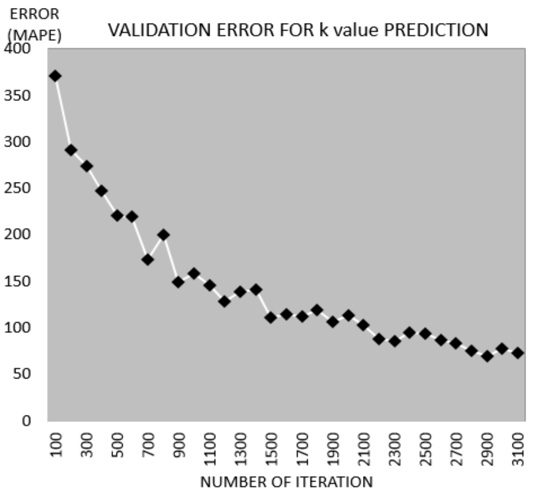

Fig. 7. Validation Errors Distribution in an Experiment

However, in a validation test, the overall trend of error values decreases but the errors fluctuate at some points. For example, error value became larger than the previous iteration for 800, 1300, and 1400 iteration. The reason for this is that over-fitting occurred during neural network training. Over-fitting means that a trained neural network is too closely fitted to the test data.

4. Conclusion

In this work, the ANN model has been defined and implemented using Tensorflow libraries to estimate a *k* value in a k-anonymity algorithm. Various models has been evaluated from the aspect of RMSE errors and processing time, and one model has been chosen as target model. The chosen model has been trained and tested using a dataset that has three features. The experiment results shows that the trend of the training errors follows a typical pattern in ANN, which is an exponential decrease. When the number of iteration reaches about 10,000, the error value becomes stable. For the validation errors, our model represents a different pattern from ones shown in training process. In validation test, the overall trend of losses decreases but those fluctuates at some points. For example, the validation losses became larger than those of the previous iteration at the points of 800, 1300, and 1400 iteration. The reason for this is because over-fitting occurred during neural network training. Therefore, to achieve better estimation of k value, training should be stopped around the point of 800 iterations.

References

[1] L. M. Kumbayoni and S.-B. Jang, "A Determination of k-value Based on Artificial Neural Network in k-anonymization," *Proceedings of International Conference on Convergence Research,* Vol.4, No.3, pp.437-440, 2018.

[2] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge based Systems*, pp.557-570, 2002.

[3] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-Preserving Data Publishing: A Survey of Recent Developments," *ACM Computing Survey*, Vol.42, No.4, pp.1-53, 2010.

[4] I. Roy, S. T. Setty, A Kilzer, V Shmatikov, and E Witchel, "Airavat: Security and Privacy for Mapreduce," *Proceedings of Seventh USENIX Conference on Networked Systems Design and Implementation*, pp.297-312, 2010.

[5] N. Elanshekhar and R. Shedge, "An effective anonymization technique of big data using suppression slicing method," *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pp.2500-2504, 2017,

[6] M. Nassar, A. A. R. Orabi, M. Doha, and B. Al Bouna, "An SQL-like query tool for data anonymization and outsourcing," *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, pp.1-3, 2015.

[7] R. Matsunaga, I. Ricarte, T. Basso, and R. Moraes, "Towards an Ontology-Based definition of Data Anonymization Policy for Cloud Computing and Big Data," *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp.75-82, 2017.

[8] X. Zhang, W. Dou, J. Pei, S. Nepal, C. Yang, C. Liu, and J. Chen, "Proximity-Aware Local-Recoding Anonymization with MapReduce for Scalable Big Data Privacy Preservation in Cloud," *IEEE Transactions on Computers*, Vol.64, No.8, pp.2293-2307, 2015.

[9] A. A. William, "On the Efficiency of Learning Machines," IEEE Transactions on Systems Science and Cybernetics, Vol.3, No.2, pp.111-116, 1967.

[10] V. F. Nicholas, "Some New Approaches to Machine Learning," *IEEE Transactions on Systems Science and Cybernetics*, Vol.5, No.3, pp.173-182, 1969.

[11] H. C. Anderson, "Neural network machines", *IEEE Potentials*, Vol.8, No.1, pp.13-16, 1989.

[12] J. Nagumo and A. Noda, "A learning method for system identification," *IEEE Transactions on Automatic Control*, Vol.12, No.3, pp.282-287, 1967.

[13] W. T. Illingworth, "Beginner's guide to neural networks," *IEEE Aerospace and Electronic Systems Magazine*, Vol.4, No.9, pp.234-241, 1989.

[14] D. Burshtein, "Long-term attraction in higher order neural networks," *IEEE Transactions on Neural Networks*, Vol.9, No.1, pp.42-50, 1998.

[15] P. D. Wasserman and T. Schwartz, "Neural networks. II. What are they and why is everybody so interested in them now?," *IEEE Expert*, Vol.3, No.1, pp.10-15, 1988.

[16] P. Yugowati, M. Shaou-Gang, and W. Hui-Ming, "Supervised learning approaches and feature selection-a case study in diabetes," *International Journal of Data Analysis Techniques and Strategies*, Vol. 5, No.3, 2013, pp. 323-337.

[17] J.-X. Peng, L. Kang, and W. I. George, "A New Jacobian Matrix for Optimal Learning of Single-Layer Neural Networks," *IEEE Transactions on Neural Networks*, Vol.19, No.1, pp.119-129, 2008.

[18] C. Gustavo, B. C. Antoni, J. M. Pedro, and V. Nuno, "Supervised Learning of Semantic Classes for Image Annotation and Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.29, No.3, pp.394-410, 2007.

Kumbayoni Lalu Muh

https://orcid.org/0000-0002-2342-4645

e-mail : kumbayoni@kumoh.ac.kr

He graduated from an department of automotive engineering in Daegu Catholic University in 2017. He is pursuing a master degree in Kumoh National Institute of Technology. His research interests include machine learning, Image recognition.

Sung-Bong Jang

https://orcid.org/0000-0003-3187-6585

e-mail : sungbong.jang@kumoh.ac.kr

He received his B.S., M.S., and Ph.D. degrees from Korea University, Seoul, Korea in 1997, 1999, and 2010, respectively. He worked at the Mobile Handset R&D Center, LG Electronics from 1999 to 2012. Currently, he is an associate professor in the Department of Industry-Academy, Kumoh National Institute of Technology in Korea. His interests include Augmented Reality, Big Data Privacy, Prediction based on Artificial Neural Networks.