Plate Recognition Using Backpropagation Neural Network and Genetic Algorithm

Conference Paper *in* Procedia Computer Science · January 2017

DOI: 10.1016/j.procs.2017.10.068

Joseph Tarigan
Binus University

Yaya Suryana
Badan Pengkajian dan Penerapan Teknologi

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 116 (2017) 365–372

Procedia Computer Science

www.elsevier.com/locate/procedia

2nd International Conference on Computer Science and Computational Intelligence 2017, ICCSCI 2017, 13-14 October 2017, Bali, Indonesia

Plate Recognition Using Backpropagation Neural Network and Genetic Algorithm

Joseph Tarigan[a], Nadia[b], Ryanda Diedan[c], Yaya Suryana[d]

[a,b,c]*Bina Nusantara University, Jl. Kebon Jeruk Raya No. 27. Kebon Jeruk, Jakarta Barat 11530, Indonesia*
[d]*BPPT, Jl. MH.Thamrin No.8, Jakarta 10340, Indonesia*

Abstract

Plate recognizer system is an important system. It can be used for automatic parking gate or automatic ticketing system. The purpose of this study is to determine the effectiveness of Genetic Algorithms (GA) in optimizing the number of hidden neurons, learning rate and momentum rate on Backpropagation Neural Network (BPNN) that is applied to the Automatic Plate Number Recognizer (APNR). Research done by building a GA optimized BPNN (GABPNN) and APNR system using image processing methods, including grayscale conversion, top-hat transformation, binary morphological, Otsu threshold and binary image projection. The tests conducted with backpropagation training and recognition test. The result shows that GA optimized backpropagation neural network requires 2230 epochs in the training process to be convergent, which is 36.83% faster than non-optimal backpropagation neural network, while the accuracy is 1,35% better than non-optimized backpropagation neural network.

Keywords: Bacpropagation Neural Network; Genetic Algorithm; Optical Character Recognition; Computer Vision; Top-Hat Transformation

1. Introduction

In the recent years, automated systems have become an integral part of daily tasks that only a human can do before. Automated systems are meant to help human to do task that involves knowledge, reasoning and experience. The integral part of an automated system is artificial intelligence and one of the application of artificial intelligence in automated system is Optical Character Recognition (OCR). OCR let a computer recognize character through visual interpretation and recognize character automatically without help from human. There are several algorithms that we can use to create OCR system, such as template matching, support vector machine (SVM), hidden markov model, hausdorff distance and artificial neural network. Artificial neural network is the most popular algorithm that has been

* Corresponding author. *E-mail address:* joseph.tarigan@binus.ac.id, nadia@binus.edu, spyro.blast3@gmail.com, yaya.suryana@gmail.com

1877-0509
10.1016/j.procs.2017.10.068

used by researcher to solve pattern recognition problems[1]. Artificial neural network can be used to solve many problems and it can be trained over time to gain its knowledge or to enhance its accuracy in recognizing patterns.

Artificial neural network is an abstraction of biological neural network that simulate the way it works in biological brain. Neurons are interconnected by synapsis that carries information and it can be modified by training process. There are several training processes that can be used to train artificial neural network, one of them is backpropagation training method. Backpropagation training method involves feedforward of the input training pattern, calculation and backpropagation of error, and adjustment of the weights in synapses.

2. Background

There are two variables that affect the speed of training process and the accuracy of pattern recognition, those are learning rate and momentum rate. Learning rate and momentum rate values that are optimal in a backpropagation neural network topology might not be optimal in another backpropagation neural network topology because each topology is unique to its usage domain. Those two variables also affect the tendency of backpropagation process fall into local minima that makes the artificial neural network recognition performance becomes sub-optimal[2].

There are several methods that can be used for a priori determining learning rate and momentum rate, those are trial-and-error and second-order method. Trial-and-error method needs human interaction; therefore, it is time consuming and needs effort. Second-order method is a method that can adjust learning rate using information gathered from training process. This method is rarely used because it needs massive computation resources[3]. The learning rate that is too high will lead into overcorrecting problem in the gradient descent process, yielding sub-optimal accuracy and longer training process[4].

Beside the learning rate and the momentum rate, the number of neuron in the hidden layer also affects the speed of backpropagation process and its accuracy. Logically, the more neuron it has, the more information can be stored, but it affects the speed of the training process. A non-optimal number of neuron also leads into a problem. Too many hidden neuron will lead into overfitting problem. In contrast, if there is only a few neurons, it will lead into under fitting problem. Both will make the artificial neural network cannot generalize the input, yielding sub-optimal performance[5, 6].

There are several methods for determining the number of neuron and layer in an artificial neural network, they are rule-of-thumb and structured trial-and-error. Rule-of-thumb method is a compilation of rules that concluded by researchers. Rule-of-thumb method doesn't guarantee the optimality of neural network because the activation function and training algorithm also affect the determination of neuron's number. Structured trial-and-error is a naïve programming method to determine the number of neuron and layer. This method will generate random number for neuron and layer number and train the neural network by trial and error. If the training process fails to converge, then the number of the neuron and layer will be increased and the neural network will be trained again. This iteration goes on up until the neural network converges[6].

In order to build an optimal artificial neural network, an algorithm that can optimize learning rate, momentum rate, and count of hidden neuron simultaneously is needed. This research introduces an approach in optimizing those variables. The algorithm that we use is genetic algorithm. Genetic algorithm mimics the natural selection. An individual will be eliminated naturally if it cannot adapt to the environment. That individual will be replaced with a new individual from crossover and mutation process. Genetic algorithm is an optimization technique, so that it can be used to find the most optimal solution from available solutions[7]. The combination of genetic algorithm and backpropagation algorithm enable the backpropagation neural network to have better accuracy, speed up the recognition and the training process rather than a backpropagation neural network without optimization process which leads to efficiency of CPU time.

3. Methodology

Genetic algorithm was used to define the optimal learning rate, momentum rate and the number of neuron in the hidden layer of a backpropagation neural network. Then the backpropagation neural network is applied in plate number recognition system. Valid Indonesia plates that comply with the regulations and with black background were used as our sample. See Fig. 1.

Fig. 1. (left) An example of a valid number plate sample, (middle) gray scaled image and (right) resulted image form top-hat transformation.

Image preprocessing is utilized to process the image. The input image will be processed by the preprocessing stage and the output will be recognized by the backpropagation neural network. Each stage will be described below.

4. Pre-processing

This step is used to locate the plate in the given photo and to locate each character in the plate. This block diagram of the steps can be seen in the diagram below.

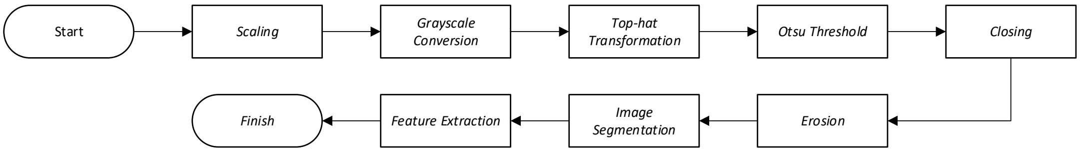

Fig. 2. Pre-processing diagram.

First, every input image will be scaled. The height of the image will be scaled into 768 pixels, and the width will be scaled according to its ratio. We convert the image into greyscale image by multiply each value in each channel with a constant and summarize the result[8]. We use the formula as follows:

$$I_y = 0.30\ f_R + 0.59\ f_G + 0.11\ f_B \quad (1)$$

Next, the unwanted area such as reflection from the number plate is eliminated and the threshold function is optimized. Top-hat transformation[12] was utilized. First, the size of structuring element was defined. If it is too small, then the character that is needed might be eliminated. If it is too big, then the unwanted area will not be eliminated. Referring to the scaling step, the size of the structuring element is defined by trail-and-error method. The rectangular structuring element with size 41 by 27 pixels was used. Next, Otsu threshold was applied to the image. This step will transform the greyscale image into binary image and will separate the character candidates which have white color from background which have black color. This conversion also eliminates some unwanted noise from the image. Otsu threshold[11] was used because it can tolerate the variation of brightness.

Next, binary closing was applied to the image. This process will close or fill the blank area inside the character that occurred because of scratch of defect. 2x2 pixels rectangular structuring element was used. Then, erosion was applied to the image. This process will eliminate small impulse noise and erode the character area. Erosion was applied because character area in the resulting image from closing process is relatively too bold. 3x3 pixels rectangular structuring element was used. Next, segmentation was done to the resulted image. In this step, each of the character candidates was segmented. There are four steps in this process, horizontal character segmentation, connected component labelling, verification and scaling.

The first step was the horizontal character segmentation. This step is done for localize the number plate area. First, vertical projection onto the image was applied, then the vertical projection was multiplied with a constant L.

$$L = 0.09 * \max(VP(image)) \quad (2)$$

The result of multiplication for vertical projection is shown in Fig. 3.

Fig. 3. Graph of vertical projection after multiplication.

Then the most vertically wide area was taken as the location of the number plate.

The next step is connected component labelling, row-by-row union labelling method was used. The image was scaled into 500 by 50 pixels width before the process to speed up the process. Then each of the labelled area was segmented using its horizontal and vertical projection data.

It is needed to verify each of the resulted images from preceding process. It was done by using minimum width and height verification and height-to-width ratio verification. The last step is to scale the resulting image. It scaled into a size that corresponds with the number of input neuron in the backpropagation neural network. It was scaled into 9 by 12 pixels image, therefore it corresponds to 108 neurons in the input layer.

Fig. 4. (left) Vertical & horizontal segmentation result. (right) Verification and scaling result.

Binary image has its pixel value as a feature. An 8-bit binary image has pixel value from 0 to 255. It is needed to be normalized first, so the backpropagation neural network can take it as an input. The normalization was done by dividing each pixel value by 255. The resulting pixel value is now ranged between 0 and 1.

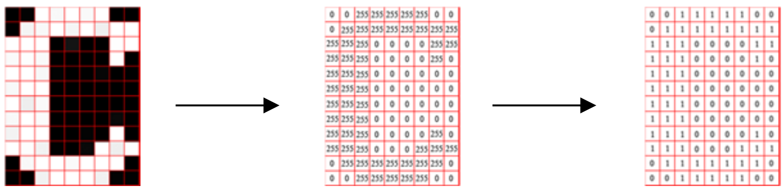

Fig. 5. Feature extraction process.

5. Backpropagation Neural Network

In this research, neural network is built after the genetic algorithm process has been done. Beside the optimization process, some parameters are still needed to be defined. Those are layer configuration, weight initiation, activation function, and backpropagation. The description is described in each section below.

5.1. Layer Configuration & Weight Initialization

In this research, 3 layered artificial neural network was utilized. It consists of input layer, hidden layer and output layer. Input layer has 109 neurons, consists of 108 input neurons that represent the number of pixel from input image and 1 bias neuron. The hidden layer will take the optimized number of hidden neuron taken from the result of genetic algorithm optimization process. The output layer has 36 neurons, each of them corresponds to alphanumeric character, A to Z and 0 to 9. Nguyen-Widrow[9] initialization method was used for the weights between input layer and hidden layer, because it makes the training process faster. For the weights between hidden layer and output layer, random initialization method was used with value from -0.5 to 0.5.

5.2. Activation Function & Backpropagation Training

Two most common activation functions that can be used are binary sigmoid and bipolar sigmoid[10]. In this research, binary sigmoid was used as the activation function, and then ten sets of alphanumeric characters were used for the training purposes. The samples consist of good samples and samples with noise. The training sets were deliberately mixed so that the neural network will have a better generalization. The neural network was trained until the mean squared error reaches the value of 0.00001.

6. Genetic Algorithm

In this research, genetic algorithm is used to optimize learning rate, momentum rate, and number of hidden neuron simultaneously. The backpropagation training was done until it reached 1600 epoch for each individual using properties gathered from the individual's genotype. There are several steps in which it is needed to be defined first. The description are as follows.

6.1. Representation, Population & Initialization

The phenotypes that make up the individual are learning rate, momentum rate, and hidden neuron number. Hidden neuron number is an integer while learning rate and momentum rate are real number ranged from 0 to 1. It is needed to convert them to genotype so they can be used in the genetic algorithm optimization process. The phenotypes were converted into binary representation. It is concluded that the number of hidden neuron should be 2/3 of its input neurons and if necessary can be added with the number of output neurons[6]. From that conclusion, the genotype of hidden neuron number is an 8-bit binary representation that represents number from 1-256.

Learning rate and momentum rate are real number and have the same binary representation. 9-bit binary representation was used as the genotype of these variables that represents real number from 0 to 1. It is concluded by Wilson and Martinez[4] that the best learning rate for a neural network to generalize lies bellow 0.005. The encoding & decoding process for learning rate and momentum rate is described as follows.

$$genotype = binaryOf(phenotype * 512) \quad (3)$$

$$phenotype = \frac{decimalOf(genotype)+1}{512} \quad (4)$$

With those formula, the search space for learning rate and momentum rate is between 1/512 and 512/512 or between 0.001953125 and 1. Combining the genotypes, each individual has 26-bit binary string which consist of 8-bit genotype for hidden layer number, 9-bit genotype for learning rate and 9-bit genotype for momentum rate.

1	0	0	1	1	1	0	1	1	0	1	1	1	0	1	0	0	0	0	1	0	1	1	0	0	1

Fig. 6. Genotype representation, blue represents hidden neuron count, green represents learning rate and orange represents momentum rate.

In this research, steady-state model was used. The population is limited to 50 individuals with mating pool that can hold 10 individuals. The initialization of each binary representation was done using random initialization with binary number, 0 and 1 for each gene.

6.2. Fitness function

The fitness function is used to evaluate the individuals in a population. The previous researchers[4] used mean square error at a particular epoch as the fitness function. In this research, mean squared error cannot be used because the initial weights are stochastic, therefore there is a possibility that the mean square error in that particular epoch is local maxima. Instead, the average of mean square error summation at a particular epoch is used. The neural network was trained up until 1600 epoch. The formula is given below.

$$fitness_i = \max(fitness) - \frac{\sum_{j=1}^{1600} MSE_j}{1600} \quad (5)$$

6.3. Parent & Survivor Selection

The roulette wheel selection was used as the parent selection mechanism. First, the individuals were sorted by their proportional fitness. Proportional fitness is calculated with the formula as follows:

$$fitness_i = \frac{fitness_i}{\max(fitness)} \quad (6)$$

Now, each of the individual has a normalized fitness value, from 0 to 1. Then random number will be generated ranged from 0 to 1 and select each individual that has the closest fitness value with the random number. This step was repeated 10 times as the mating pool capacity is 10 individuals. Selection by fitness was used as the survivor selection. In each varian operation, offsprings are joined with individuals in the population, then it was sorted descending by their fitness. Ten lowest individuals will be eliminated and the rest will be used as the population in the next generation.

6.4. Crossover, Mutation & Termination condition

In this research, uniform crossover was used. A pair of individuals was selected randomly from the mating pool. The crossover probability in this research is 50%, then a random number was generated in the range from 0 to 1. If the random number exceeds the crossover probability, which is 0.5, then the crossover was done on that pair. For the mutation, bit flip mutation was used. The probability of mutation in each gene is 10%. Bit-flip is done by flipping the binary number in the gene. For the termination, the termination condition is the number of generation. The number of the generation is limited to 150. The genetic algorithm process will stop when it reaches the 150th generation, then the parameters from an individual with the highest fitness were used to build the backpropagation neural network.

7. Results & Discussion

The genetic algorithm optimization process ran for 26 hours, 37 minutes and 6 seconds with the maximum fitness value in the last generation was 10.181856. The genotypes from that individual was encoded, the result is presented in the table below.

Table 1. The decoded genotypes

	Genotype	Phenotype
Hidden Neuron	11111100	252
Learning Rate	000011001	0.050781
Momentum Rate	111111000	0.986328

The phenotypes were used to build and train the backpropagation neural network. In order to compare the difference between an ordinary backpropagation neural network and an optimized backpropagation neural network, another backpropagation neural network was built for comparative purpose. The number of the hidden neuron were determined using rule-of thumb[6] and the learning rate was set to 0.2 and momentum rate was set to 0.9. The detail of the backpropagation neural network that we use as the comparator is shown below.

Table 2. Comparator backpropagation neural network

	Value
Hidden Neuron	72
Learning Rate	0.2
Momentum Rate	0.9

7.1. Backpropagation training test

The backpropagation training on both neural networks were done 10 times and the data were collected. The data are shown below.

Table 3. Backpropagation training result

No	Epoch		Time (Second)	
	GABPNN	BPNN	GABPNN	BPNN
1.	2370	3960	200	243
2.	2280	3290	200	205
3.	2070	3540	201	222
4.	2550	4010	222	259
5.	2230	3690	189	237
6.	2340	3920	205	244
7.	2270	3780	199	238
8.	2260	3740	197	238
9.	2240	3530	214	223
10.	2450	3790	212	240

One GABPNN and one BPNN will be selected out from the samples above. The selection space of each type need to be defined by specifying its boundary. It is done by using the average and standard deviation of epoch. The minimum and the maximum boundary of the selection space were defined by using these formula as follows:

$$Minimum_{Selection\ Space} = AVG_{epoch} - SD_{epoch} \quad \text{..............................} (7)$$
$$Maximum_{Selection\ Space} = AVG_{epoch} + SD_{epoch} \quad \text{..............................} (8)$$

The neural network that has the smallest epoch number between the minimum and the maximum boundary of the selection space from each type was selected. By using that method, the 5th neural network for GABPNN and 9th neural network for BPNN were selected.

Table 4. Selected backpropagation neural networks

	GABPNN	BPNN
Epoch	2230	3530
Training Time	3 Minutes 9 Seconds	3 Minutes 43 Seconds

Below is a diagram that shows the mean square error in the first 300 epochs in backpropagation training process of both neural network.

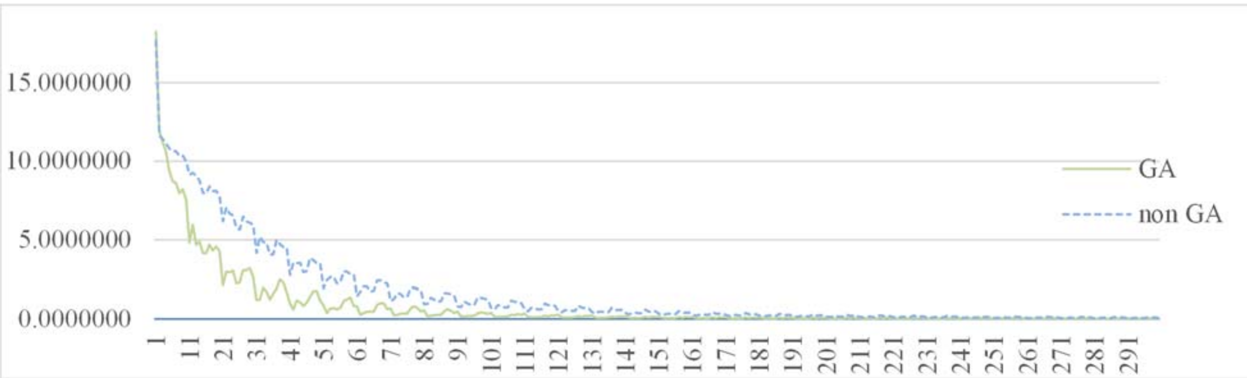

Fig. 7. MSE of the first 300 epochs in the backpropagation training.

7.2. Recognition test

The representation of each backpropagation neural network has been found. Then, recognition tests using both backpropagation neural network were carried out. As we couldn't find Indonesian Number Plate benchmark dataset, 220 photographs of Indonesian Number Plate were taken from Bina Nusantara Anggrek campus parking area. We only took photographs of standard issue number plate, according to letter of national police number ST/810/IV/2011 with black background colour. 220 photos were taken with various lighting condition and plate condition as the samples for this recognition test. Single character recognition accuracy, whole character recognition accuracy, feedforward time and average pre-processing time were collected. The collected data is shown below.

Table 5. Recognition test result of GABPNN & BPNN

Data	Result	
	GABPNN	BPNN
Single character recognition accuracy	97,18%	96,94%
Average pre-processing time	229,88 ms	229,17 ms
Average feedforward time	2,50 ms	1,38 ms
Whole characters recognition accuracy	85,97%	84,62%

8. Conclusion

In this paper, genetic algorithm was used to find the optimal number of hidden neuron, learning rate, and momentum rate. GABPNN has faster training time and require less epoch in the training process to reach the designated mean square error than BPNN, therefore, it needs less CPU time in the training process, leads to efficiency of CPU time. GABPNN also has a better recognition accuracy. Despite of the speed, it has slower feedforward time due to the number of hidden neuron.

Further research is needed to determine which variables of the BPNN or combination of those variables should be optimized and what methods and operations should be used in the GA process to find better and optimal result. We suggest the researchers to implement more robust pre-processing methods that can locate the number plate adaptively.

References

1. Anagnostopoulos CN, Anagnostopoulos IE, Loumos V, Kayafas E. A license plate-recognition algorithm for intelligent transportation system applications. IEEE Transactions on Intelligent transportation systems. 2006 Sep;7(3):377-92.

2. Haykin SS, Haykin SS, Haykin SS, Haykin SS. Neural networks and learning machines. Upper Saddle River, NJ, USA: Pearson; 2009.

3. Kandil N, Khorasani K, Patel RV, Sood VK. Optimum learning rate for backpropagation neural networks. In Electrical and Computer Engineering, 1993. Canadian Conference on 1993 Sep 14 (pp. 465-468). IEEE.

4. Wilson DR, Martinez TR. The need for small learning rates on large problems. In Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on 2001 (Vol. 1, pp. 115-119). IEEE.

5. Piotrowski AP, Napiorkowski JJ. A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling. Journal of Hydrology. 2013 Jan 7;476:97-111.

6. Karsoliya S. Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture. International Journal of Engineering Trends and Technology. 2012 Dec;3(6):714-7.

7. Eiben AE, Smith JE. Introduction to evolutionary computing. Heidelberg: springer; 2003 Nov 1.

8. Wang M, Lai CH. A concise introduction to image processing using C++. CRC Press; 2016 Apr 19.

9. Nguyen D, Widrow B. Improving the learning speed of 2-layer neural neural network by choosing initial values of the adaptive weights. In Proceedings of IEEE International Joint Conference on Neural Networks 1992 (Vol. 5, No. 4, pp. 595-603).

10. Karlik B, Olgac AV. Performance analysis of various activation functions in generalized MLP architectures of neural networks. International Journal of Artificial Intelligence and Expert Systems. 2011;1(4):111-22.

11. Otsu N. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics. 1979 Jan;9(1):62-6.

12. Gonzalez RC, Woods RE. Digital Image Processing. Pearson Education. New Jersey. 2008.