Dynamic Neuro-genetic Weights Connection Strategy for Isolated Spoken Malay Speech Recognition System

Noraini Seman, Zainab Abu Bakar, and Nordin Abu Bakar

Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA (UiTM), 40450 Shah Alam,
Selangor, Malaysia
{aini,zainab,nordin}@tmsk.uitm.edu.my

Abstract. This paper presents the fusion of artificial intelligence (AI) learning algorithms that combined genetic algorithms (GA) and neural network (NN) methods. These both methods were used to find the optimum weights for the hidden and output layers of feed-forward artificial neural network (ANN) model. Both algorithms are the separate modules and we proposed dynamic connection strategy for combining both algorithms to improve the recognition performance for isolated spoken Malay speech recognition. There are two different GA techniques used in this research, one is standard GA and slightly different technique from standard GA also has been proposed. Thus, from the results, it was observed that the performance of proposed GA algorithm while combined with NN shows better than standard GA and NN models alone. Integrating the GA with feed-forward network can improve mean square error (MSE) performance and with good connection strategy by this two stage training scheme, the recognition rate can be increased up to 99%.

Keywords: Artificial Neural Network, Conjugate Gradient, Genetic Algorithm, Global Optima, Feed-forward Network.

1 Introduction

Speech is the most natural way of communication for humans. The aim of speech recognition is to create machines that are capable of receiving speech from humans (or some spoken commands) and taking action upon this spoken information [1]. Although it was once thought to be a straightforward problem, many decades of research has revealed the fact that speech recognition is a rather difficult task to achieve, with several dimensions of difficulty due to the non-stationary nature of speech, the vocabulary size, speaker dependency issues, etc. [1]. However, there have been quite remarkable advances and many successful applications in speech recognition field, especially with the advances in computing technology beginning in the 1980s.

In recent years, there has been an increasing interest in classification approach to improvements the recognition of speech sounds. Various approaches have been made up to develop the speech recognizer or classifier and over the years there are three

V.V. Das and P.M. El-Kafrawy (Eds.): SPIT 2012, LNICST 117, pp. 121–130, 2014.

speech recognition approaches that have been developed. Dynamic time warping (DTW) is the oldest approach and is an algorithm for measuring similarity between two sequences which may vary in time or speed [2][3]. However, this technology has been displaced by the more accurate Hidden Markov Model (HMM) that has become the primary tool for speech recognition since the 1970s. Hidden Markov Model (HMM) is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters. This algorithm is often used due to its simplicity and feasibility of use.

However in late 1980s, artificial intelligent (AI) based approaches are considered for training the system to recognize speech using an artificial neural network (ANN) algorithms. This technology is capable of solving much more complicated recognition tasks, and can handle low quality, noisy data, and speaker independence. Researchers have started to consider the ANN as an alternative to the HMM approach in speech recognition due to two broad reasons: speech recognition can basically be viewed as a pattern classification problem, and ANN can perform complex classification tasks [4]. Given sufficient input-output data, ANN is able to approximate any continuous function to arbitrary accuracy.

However, the main obstacles that faced by NN model is a longer learning time when the data set becomes greater. NN learning is highly important and is undergoing intense research in both biological and artificial networks. A learning algorithm is the heart of the NN based system. Error Back-Propagation (EBP) [5] is the most cited learning algorithm and yet powerful method to train ANN model [6]. However, there are several drawbacks in the EBP learning algorithms; where the main basic defect is the convergence of EBP algorithms which are generally slow since it is based on *gradient descent* minimization method. Gradient search techniques tend to get trapped at local minima.

Recently, many researchers tried to overcome this problem by using the stochastic algorithm, such as Genetic Algorithms (GA) [7], since they are less sensitive to local minima. Genetic Algorithm (GA) based learning provides an alternative way to learn for the ANN, which involves controlling the learning complexity by adjusting the number of weights of the ANN. However, GA is generally slow compared to the fastest versions of gradient-based algorithms due to its nature to find a global solution in the search space. Thus, to have better time to converge, it is a good idea to combine the global search GA method with matrix solution second order gradient based learning methods known as Conjugate Gradient (CG) method to find the optimal values for the weights in two-layer Feed-Forward NN architecture. Therefore, we proposed fusion techniques of artificial intelligence (AI) algorithm which combines GA in the first layer and CG in the second layer to achieve optimum weights for FF network. Our algorithm aims to combine the capacity of GA and CG in avoiding local minima and the fast execution of the NN learning algorithm.

In this work, the GA-ANN model is used for validation recognition performances of isolated spoken Malay utterances and evolution in network connection weights using GA will be highlighted. Malay language is a branch of the Austronesian (Malayo-Polynesian) language family, spoken as a native language by more than 33,000,000 persons distributed over the Malay Peninsula, Sumatra, Borneo, and

numerous smaller islands of the area and widely used in Malaysia and Indonesia as a second language [8]. The direction of this work is composed into several sections, where Section 2, will explain the Malay speech materials. The details of the methods and implementation of the methods will be described in Section 3. Section 4 describes the results and discussions on the experimental of the training and validation approaches. Lastly, in Section 5, the paper is ended with conclusions.

2 Speech Collection

All experiments are conducted on the whole *hansard* document of Malaysian House of Parliament that consists of spontaneous and formal speeches. *Hansard* document is the daily record of the words spoken in the hearings of parliamentary committees. *Hansard* is not a verbatim (word for word) record of parliamentary business but is a useful record that enables interested people to check what members and senators have said and what witnesses have told parliamentary committees. The document comprises of live video and audio recorded that consists of disturbance and interruption of speakers, and contain noisy environment from different kind of speakers (Malay, Chinese and Indian). The reason of choosing this kind of data is due to its natural and spontaneous speaking styles during each session.

The most frequently words used during eight hours of one day Parliament session are determined. After some analysis, the quantitative information shows that only 50 words that most commonly used by the speakers with more than 25 repetitions. The selection of 50 words are the root words that formed by joining one or two syllables structures (CV/VC – consonant or vowel structure) that can be pronounced exactly as it is written and can control the distribution of the Malay language vocalic segments. However, the vocabulary used in this study consisted of seven words as given in Table 1, due to different selection according to their groups of syllable structure with maximum 25 repetitions and spoken by 20 speakers. Thus, the speech data set consists of 3500 utterances of isolated Malay spoken words. For the experiments, all the audio files were re-sampled at a sampling rate of 16 kHz, where the frame size is 256 kbps. All the signals data will be converted into a form that is suitable for further computer processing and analysis.

Table 1. Selected Malay words as speech target sounds

Words	Structures	Occurrences
ADA (*have*)	V + CV	3037
BOLEH (*can*)	CV + CVC	5684
DENGAN (*with*)	CV + CCVC	7433
IALAH (*is*)	VV + CVC	4652
SAYA (*i*)	CV + CV	6763
UNTUK (*for*)	CV + CVC	4101
YANG (*that*)	CVCC	4718

3 Methods and Implementation

The general idea towards this work is to generate a speech recognizer for isolated spoken Malay utterances by implementing genetic algorithm (GA) with Artificial Neural Network (ANN) to determine the suitable network architecture and to improve the recognition performance in an offline mode. The overall process of this model is described as a block diagram as shown in Fig. 1 below.

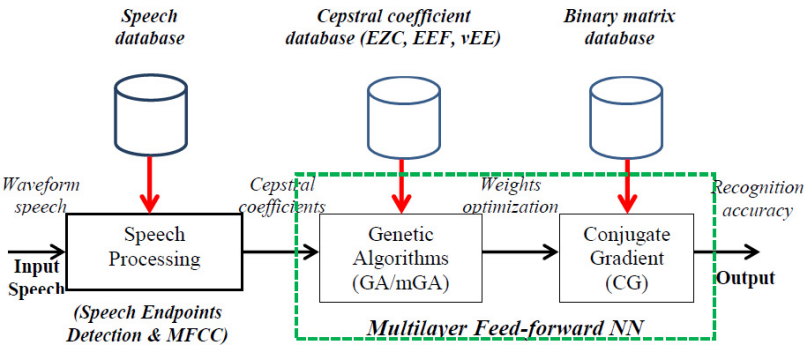

Fig. 1. Block diagram of the isolated spoken Malay speech recognition system

All the speech inputs will go through the first block of speech processing techniques that involved spectral analysis, speech boundary or endpoint detection methods, time axis normalization, feature extraction to form vector input signals for further analysis and recognition purposes. The pre-processing block designed in speech recognition aims towards reducing the complexity of the problem before the next stage start to work with the data.

Classification is the next step to identify input speeches based on the feature parameters. A two-layer feed-forward neural network with one hidden layer and one output layer was used in this work. Only one hidden layer was utilized as it had proved that an ANN with one hidden layer was sufficient in performing process mapping arbitrarily [9]. The approach combines genetic algorithm (GA) with matrix solution methods to achieve optimum weights for hidden and output layers. The proposed method is to apply genetic algorithm (GA) in the first layer and conjugate gradient (CG) method in the second layer of the FF ANN architecture as depicted in Fig. 2. These two methods are combined together using proposed dynamic connection strategy, where a feedback mechanism exists for both the algorithms.

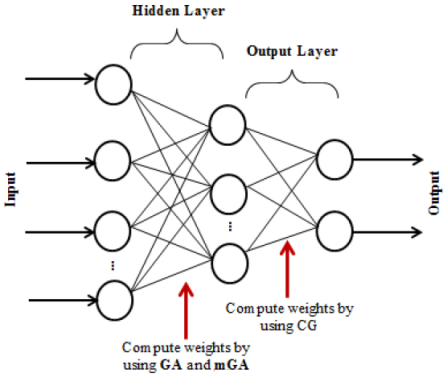

Fig. 2. The two-layer ANN architecture for the proposed weights connection of Neuro-Genetic learning algorithms

The proposed method will be compare with the standard ANN that used error back-propagation (EBP) learning algorithm. In this study, trial and error approach was used to determine the optimum topology of the network. It was found that the optimum topology of the network could be best estimated using a network with 20 hidden neurons. Using this network topology, the training and validation errors were 1.9143×10^{-5} and 1.6126×10^{-4} respectively.

In this work, we proposed two variations of genetic algorithm (GA) that can be applied for weights searching in the first layer of ANN. The first strategy is the primitive or straight GA which is applied to ANN phenotype using a direct encoding scheme and we follow exactly the work done by [10]. This GA methodology uses the standard two point crossover or interpolation as recombination operator and Gaussian noise addition as mutation operator. Meanwhile, as the second strategy, we proposed slight variations of the standard GA that is used for testing and known as **m**utation **G**enetic **A**lgorithm (**mGA**), where the only genetic operator to be considered is mutation. The mutation is applied using a variance operator. The stepwise operation for mGA can be described as follows:

Step 1: Uniform distribution technique will be used to initialize all the hidden layer weights of a closed interval range of [-1, +1]. A sample genotype for the lower half gene from the population pool for input (n), hidden units (h), output (m) and number of patterns (p) can be written as in Equation (1).

$$\begin{vmatrix} x_{11}\mu_{11}x_{12}\mu_{12}\ldots x_{1n}\mu_{1n}x_{21}\mu_{21}x_{22}\mu_{22}\ldots \\ x_{2n}\mu_{2n}\ldots x_{h1}\mu_{h1}x_{h2}\mu_{h2}\ldots x_{hn}\mu_{hn} \end{vmatrix} \quad (1)$$

where, range (x) initially is set between the closed interval [-1, +1]. Each values of variance vectors (μ) is initialized by a Gaussian distribution of mean (0) and standard deviation (1).

Step 2: The fitness for the population is calculated based on the phenotype and the target for the ANN.

$$netOutput = f(hid * weight) \tag{2}$$

where, *hid* is the output matrix from the hidden layer neurons, *weight* is the weight matrix output neurons and f is the sigmoid function is computed as in Equation (3) and (4).

$$RMSError = \sqrt{\frac{\sum_{i=1}^{n}(netOutput - net)^2}{n * p}} \tag{3}$$

$$popRMSError_i = norm(RMSError_i) \tag{4}$$

Step 3: Each individual population vector ($\mathbf{w_i}$, $\boldsymbol{\eta_i}$), i =1, 2,…, μ creates a single offspring vector ($\mathbf{w_i'}$, $\boldsymbol{\eta_i'}$) for $j = 1,2,...,n$ as in Equation (5) and (6).

$$\eta_i'(j) = \eta_i(j)\exp(\tau' N(0,1) + \tau Nj(0,1)) \tag{5}$$

$$w_i'(j) = w_i(j) + \eta_i'(j)Nj(0,1) \tag{6}$$

Step 4: Repeat **step 2**, if the convergence for the mGA is not satisfied.

Meanwhile, the weights for the output layer is computed using the conjugate gradient (CG) method where the output of the hidden layer is computed as sigmoid function [f(.)] for the weighted sum of its input. The CG algorithm is a numerical optimization technique designed to speed up the convergence of the back-propagation algorithm. It is in essence a line search technique along any set of conjugate directions, instead of along the negative gradient direction as is done in the steepest descent approach. The power of the CG algorithm comes from the fact that it avoids the calculation of the Hessian matrix or second order derivatives, yet it still converges to the exact minimum of a quadratic function with *n* parameters in at most *n* steps [11]. The conjugate gradient algorithm starts by selecting the initial search direction as the negative of the gradient as in Equation (7) and (8).

$$\underline{p}_0 = -\underline{g}_0 \tag{7}$$

$$\underline{g}_i = \nabla \underline{F}(\underline{x})|_{x=x_k} \tag{8}$$

where $\underline{x}$ is the vector containing the weights and biases and $\underline{F}(\underline{x})$ is the performance function, that is the mean square error (MSE). The search directions ($\underline{p}_i$) are called *conjugate* with respect to a positive definite Hessian matrix if,

$$\underline{p}_i^T \underline{A} \underline{p_i} = 0 \quad \text{for } i \neq j \tag{9}$$

where $\underline{A}$ represents the Hessian matrix [$\nabla^2 \underline{F}(\underline{x})$].

The above condition can be modified to avoid the calculation of the Hessian matrix for practical purposes and is given as in Equation (10).

$$\nabla \underline{g}_i^T \underline{p}_i = 0 \text{ for } i \neq j \tag{10}$$

The new weights and biases are computed by taking a step with respect to the learning rate (α_i) along the search direction that minimizes the error as in Equation (11).

$$\underline{x}_{i+1} = \underline{x}_i + \alpha_i \underline{p}_i \tag{11}$$

where, the learning rate (α_i) for the current step is given by Equation (12).

$$\underline{p}_{i+1} = -\underline{g}_{i+1} + \beta_{i+1} \underline{p}_i \tag{12}$$

where the scalar (β_i) which can be viewed as a momentum added to the algorithm (Duda et al. 2001) is given by one of three common choices where we adopted *Fletcher and Reeves* formula for the current implementation.

$$\beta_i = \frac{\underline{g}_{i+1}^T \underline{g}_{i+1}}{\underline{g}_i^T \underline{g}_i} \tag{13}$$

The algorithm iterates along successive conjugate directions until it converges to the minimum, or a predefined error criterion is achieved. As is obvious from the above steps, the conjugate gradient algorithm requires batch mode training, where weight and bias updates are applied after the whole training set is passed through the network, since the gradient is computed as an average over the whole training set [6]. In this work, since the network architecture is a two-layer feed-forward ANN, the input nodes in the first layer will begin with the range compression for the applied input (based on pre-specified range limits) so that it is in the open interval (0,1) and transmit the result to all the nodes in the second layer, which is the hidden layer. The hidden nodes perform a weighted sum on its input and then pass through the sigmoidal activation function before sending the result to the next layer, which is the output layer. The output layers also perform the same weighted sum operation on its input and pass through the sigmoidal activation function to produce the final result. The vital and challenging task is to find suitable rules of joining two different techniques for the given ANN architecture. The combination of the GA and the CG method provides much possibilities of joining the two different methods [10].

We proposed the dynamic connection strategy for combining these two methods, where the CG method is called after one generation run for GA/mGA method. The best fitness population is halved and the upper half is saved as the weights for output layers. Then, the GA/mGA is run for the remaining generation and the flowchart of the process is illustrated as depicted in Fig. 3.

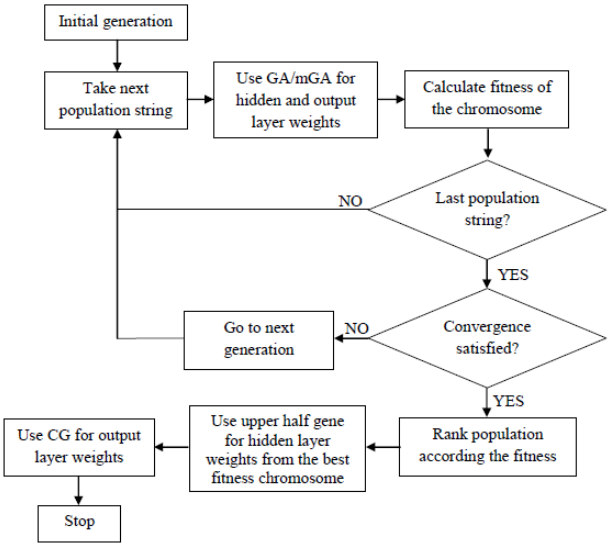

Fig. 3. The proposed dynamic weights connection strategy process diagram

4 Results and Discussion

A total of 3500 data were generated using these model inputs for modeling purposes. The data were equally divided into training and testing set. The network was obtained after undergoing a series of training using two different algorithms. In order to improve network generalization ability, early stopping techniques was applied to CG training. In this technique, validation error was monitored during the training process. When the validation error increases for a specified number of iterations, the training was stopped to prevent over fitting. For weights evolved using GA, the number of generation was used to stop the iteration.

The word recognition results obtained with the average classification rate and 95% confidence interval for the training and testing sets of all the methods used in the study is depicted in Table 2. There are 50 experiments were done to choose perfect Hidden Neurons Number (HNN) for the models. Here we specified the network configuration with the best HNN is (50-20) of multilayer feed-forward network structure.

Table 2. Confidence interval for training and testing sets with different methods

OVERALL CLASSIFICATION RATE			WORDS	DATA BELONGING TO:											
EBP	GA+CG	vGA+CG		ADA			BOLEH			DENGAN			IALAH		
93.17%	97.94%	99.17%		EBP	GA+CG	vGA+CG	EBP	GA+CG	vGA+CG	EBP	GA+CG	vGA+CG	EBP	GA+CG	vGA+CG
DATA IDENTIFIED AS:			ADA	95.88333	99.05417	99.72083	0.275167	0.045833	0.008333	3.675	1.033333	0.183333	0.25833	0.033333	0.004167
			BOLEH	1.345833	0.8375	0.075	95.47917	98.80833	99.56417	2.4625	1.279167	0.8375	2.296833	0.783333	0.304167
			DENGAN	1.9125	0.479167	0.1625	0.845833	0.283333	0.1375	89.475	96.67083	98.43333	3.670833	1.0375	0.591667
			IALAH	0.036667	0.004167	0	0.7	0.204167	0.045833	1.683333	0.625	0.208333	93.20417	98.61667	99.41667
			SAYA	0	0	0	0.908333	0.354167	0.058333	0.575	0.066667	0.008333	2.3	0.75	0.454167
			UNTUK	0.145833	0.05	0.125	1.7625	0.304167	0.095833	1.170833	0.183333	0.0375	1.33333	0.3125	0.129167
			YANG	0.695833	0.075	0.029167	0.025	0	0	0.758333	0.166667	0.066667	1.0375	0.266667	0.1

OVERALL CLASSIFICATION RATE			WORDS	DATA BELONGING TO:								
EBP	GA+CG	vGA+CG		SAYA			UNTUK			YANG		
93.17%	97.94%	99.17%		EBP	GA+CG	vGA+CG	EBP	GA+CG	vGA+CG	EBP	GA+CG	vGA+CG
DATA IDENTIFIED AS:			ADA	0.275	0.029167	0.004167	0.004167	0	0	0.579167	0.066667	0.016667
			BOLEH	2.033333	0.8675	0.245833	1.658333	0.55	0.183333	0.05	0.008333	0
			DENGAN	0.258333	0.033333	0.029167	0.075	0	0.004167	0.15	0.033333	0.0125
			IALAH	3.458333	1.65	0.725	3.325	0.875	0.308333	0.133333	0.0625	0.045833
			SAYA	89.10833	96.17917	98.525	0.591667	0.108333	0.0125	0.058333	0.0125	0
			UNTUK	3.658333	1.175	0.416667	94.20833	98.44167	99.4875	0.1	0	0
			YANG	1.208333	0.245833	0.054167	0.1375	0.025	0.004167	98.92917	99.81667	99.925

The maximum number of epochs for network training was set to 1,000 after observing that convergence was reached within the range. From the result, it shows that the best network trained using the coalition of different AI algorithms. Here, the proposed algorithm using mutation Genetic Algorithm (mGA) and Conjugate Gradient (CG) yielded 99.17% of overall classification rate. The proposed method outperformed other two training networks where 97.94% obtained from fusion of standard GA and CG, meanwhile standard ANN using EBP algorithm yielded 93.17% is the lowest among other two algorithms. Although the difference in overall classification performances between standard GA and CG (GA+CG) and the mGA with CG (mGA+CG) may seem small, the difference between the two algorithms becomes more significant when the individual confusions matrices and 95% confidence interval plots are examined.

The degradation in recognition is very noticeable on all the vocabulary words except from word “ADA” and “YANG”. The spreads in confidence intervals of the words “BOLEH” and “UNTUK” obtained with the GA+CG algorithm are 16.25% and 18.55% respectively. Whereas the spreads for the same words obtained with the mGA+CG method are 5.6% and 3.7% respectively. Therefore, the mGA+CG leads to more accurate and reliable learning algorithm to trained the FF network for this word recognition study than the standard GA+CG algorithm does.

Since the calculation started with random initial weights, each run produced different results even though the network architecture was maintained. Thus, in order to obtain an optimal solution, repeated runs were practiced and only the best result was recorded. This can be done because the convergence time of CG training method was really fast.

Owing to this fact, GA combined with CG has some given advantages. Moreover, the performances in the validation sets were considered better than standard ANN using EBP algorithm and this proved that this scheme was adequate with a sufficient accuracy.

5 Conclusions

Based on the results obtained in this study, ANN is an efficient and effective empirical modeling tool for estimating the speech process variable by using other easily available process measurements. The use of multilayer feed-forward network with delay values in model input variables is sufficient to give estimation to any arbitrary accuracy. Even though the conventional EBP method is widely used, but the GA is more preferable as the optimal solution searching is population based that using gradient information. Integrating the GA with CG as second order gradient based learning method can improve MSE performance and by this two stage training scheme, the recognition rate can be increasing up to 85%. However, speech recognition rate still has room for improvement, where much effort is needed to improve GA method for speeding up the learning process in ANN model.

References

1. Deller, J.R., Proakis, J.G., Hansen, J.H.L.: Discrete-Time Processing of Speech Signal. Macmillan, New York (1993)
2. Itakura, F.: Minimum prediction residual principle applied to speech recognition. IEEE Transactions on Acoustic, Speech and Signal Processing 1975 23(1), 67–72 (1975)
3. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustic, Speech and Signal Processing 26(1), 43–49 (1978)
4. Panayiota, P., Costa, N., Costantinos, S.P.: Classification capacity of a modular neural network implementing neurally inspired architecture and training rules. IEEE Transactions on Neural Networks 15(3), 597–612 (2004)
5. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representation by error propagation. In: Parallel Distributed Processing, Exploring the Macro Structure of Cognition. MIT Press, Cambridge (1986)
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, New York (2001)
7. Goldberg, D.E.: Genetic Algorithm in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
8. Britannica, Encyclopedia Britannica Online (2007), http://www.britannica.com/eb/article-9050292
9. Hornik, K.J., Stinchcombe, D., White, H.: Multilayer Feedforward Networks are Universal Approximators. Neural Networks 2(5), 359–366 (1989)
10. Ghosh, R., Yearwood, J., Ghosh, M., Bagirov, A.: Hybridization of neural learning algorithms using evolutionary and discrete gradient approaches. Computer Sciernce Journal 1(3), 387–394 (2005)
11. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design. University of Colorado, Campus Publishing Service (1996)