Chapter 2: Literature Review

2.1 Introduction

This chapter introduces the published knowledge in vowel recognition. It summarizes the findings of the latest studies in the related topic. Then, a brief description for the current classifiers used for vowel recognition. The architectures of FFBP and Elman NN were explained. Finally, *k*-fold cross validation technique was highlighted.

2.2 Overview

Speech recognition plays a vital role in the modern communication systems. Applications for speech recognition system vary from voice dialing and data entry to credit card identification and speech-to-text processing (Xiaoming & Baoyu, 1998). Speech recognition is simply to convert spoken words into text (Davis, Biddulfh, & Balashek, 2002). Even though there is a huge development in the field of speech recognition, still there are some technological restrictions to get user satisfactions. These limitations are due to the sensitivity to the environmental noise and weak presentation of the grammatical knowledge (Benzeghiba et al., 2007). Hence, speech recognition research area is very important to introduce more accurate speech recognizers.

Speech signal carry the linguistic information as well as the speaker's personal information (i.e. age, gender, emotional state…etc) (Nolan, 1980). Phonemes are the basic structure that differentiates meaning. In most languages (i.e. English), words are formed by combining phonemes. Normally, the consonant-vowel units are more frequent than various forms of subword units. Hence, the developing of high accuracy speech recognition

systems is highly dependable on the ability to classify consonant-vowel units with high recognition rate (Shahrul, Siraj, Yaacob, Paulraj, & Nazri, 2010).

Malay language has 24 pure Malay phonemes which consist of 18 consonants and 6 vowels. This is an advantage for the Malay language over English language which has 20 vowel phonemes (Shahrul, et al., 2010). Table 2.1 shows a list of Malay vowels according to the experiments performed by Hassan (1980) and Karim et al. (1995) .

Table 2.1: List of Malay vowels.

Tongue Position / Tongue Height	Front	Center	Back
High	i		u
Mid-hight	e	ə	o
Mid-low			
Low	a		

Vowels, in the phonetic definition, are the sounds produced without blocking the air coming from lungs, keeping the vocal track opened when they are pronounced. The word vowel was derived from the Latin word *vocalis*, which means speaking. This is because of the fact that large percentage of words in most languages contains vowels. The quality of vowels depends on the articulatory features that differentiate vowel sounds (Lingling & Kuihe, 2009).

Studies in the literature approved that there is a difference between children's speech and adult's speech in terms of absolute values and variability of linguistic and acoustic correlates (Potamianos & Narayanan, 2007). Moreover, Wilpon and Jacobsen (1996) argued that speech recognizers perform well in the age range between 15 and 70; however, the accuracy of the recognizers decreased dramatically outside this range.

Many vowel production studies found in the literature have investigated the relationship between the fundamental frequency f_o of speech and age. These studies have been confirming that children have higher f_o than adults, and speech of children contains higher pitch than speech of adults (Potamianos & Narayanan, 2003). This is true because children's vocal folds and vocal tract is still in the growing stage (Lee & Iverson, 2009). As a result, children's speech recognition is more difficult than adults' speech recognition (Giuliani & Gerosa, 2003).

2.3 Speech Recognition

Many pattern recognition methods have been implemented in field of speech recognition. One classical technique is nearest neighbor algorithm where the strength of matching is the gap between the unknown templates and the reference (Botros, 1991). However, these approaches have constrains in terms of classifying broad vocabularies in speaker-independent systems. Hidden Markov Model (HMM) is another technique for speech recognition. The main advantage of HMM is its great ability to perform segmentation in the temporal domain. However, HMMs are weak in recognition because the maximum likelihood estimation training is not discriminant (El-Ramly, Abdel-Kader, & El-Adawi, 2002). Therefore, researchers have started to focus more on NNs whose discrimination is one of its advantages.

2.3.1 Vowel Recognition

A considerable amount of the literature has been published on vowel recognition. Carlson and Glass (1992) have conducted many vowel classification experiments based on speech-synthesis-like parameters. Vuckovic and Stankovic (2001) compared different methods of the automatic vowel classification based on 2-dimensional formant Euclidean distance. Moreover, Nong et al. (2001) used multilayer perceptron and dynamic time warping techniques to classify Malay vowels.

In addition to that, formant features were investigated for classification of accents conversations between the three major English speakers (American, British and Australian) using linear prediction model feature analysis and a 2-D hidden Markov model (Qin & Vaseghi, 2003). Also, Mohammad Nazari et al.(2008) implemented a kernel-based feature extraction with SVM classifier for speaker-independent vowels recognition in Persian speech using non-linear dimension reduction techniques.

Furthermore, formant characteristics of vowels produced by mandarin esophageal speakers were studied by Liu and Ng (2009). They examined the first three formant frequencies using Praat's linear predictive coding algorithm. Recently, a new technique was suggested by Shahrul et al., (2010) for Malay vowel feature extraction using first and second formant and based on spectrum envelope called first formant bandwidth.

Table 2.2 summarizes latest studies in the area of vowel recognition (Shahrul, et al., 2010). The table shows the speaker type, frame type used in the analysis and the accuracy of recognition rate. Multi-frame analysis was used for almost all recent studies related to dependent and independent speaker systems. It is shown in the table that the accuracy for speaker-dependent recognition systems ranges between 89 to 100%. On the other hand, this range is reduced in speaker-independent recognition systems to be between 70 to 94%.

Table 2.2: Latest studies related to vowel recognition (Shahrul, et al., 2010).

Reference	Speaker Type	Frame Analysis	Accuracy (%)
Mohammad Nazari et. al., 2008	Independent	Multi-Frame	93.9
Ting & Mark, 2008	Dependent	Multi-Frame	98-100
Mara Carvalho	Dependent	Multi-Frame	89-96
Bresolin et.al, 2007	Independent/ Dependent	Multi-Frame	91.01/ 98.07
Muralishankar, 2005	Independent	Multi-Frame	71.72
Merkx and Miles, 2005	Independent	Multi-Frame	91.5
Ting & Yunus, 2004	Independent	Single-Frame	76.25

It is important to mention here that according to a practical knowledge, it has been found that the time used for training NN using multi-frame analysis is longer than that using single-frame analysis. The basic explanation for this time difference is the fact that the number of **L**inear **P**redictive **C**oding (LPC) coefficients representing each vowel signal in the multi-frame analysis is higher than that in single-frame analysis. Other factors affect the training time are the number of the neurons in the hidden layer and the training function being used.

2.4 Neural Networks (NNs)

Artificial NNs is a collection of layers of neurons. Each neuron takes an input from neurons in the previous layer, or it takes external input if the neuron in the first layer. After that, this input adds it up, and passes it to the next layer. Each connection between layers has a certain weight. NNs adjust these weights every time it processes any input; such that

it derives the output as close as possible to the given target. This process is called "training". After several training iterations, the NN can produce the correct output (Kumar, Kumar, & Rajan, 2009). Many NN architectures have been implemented; however, the current study is comparing the performance of two popular NN architectures: FFBP and Elman recurrent networks.

2.4.1 Feed-forward Back Propagation Network

FFBP network is the best network architecture among the Multi-layer Perceptron (MLP) networks (M. K. S. Alsmadi, Omar, & Noah, 2009). FFBP network uses supervised learning approach and consisted of at least three layers: input layer, hidden layer(s) and output layer. Input layer contains the source neurons which transfer the external input signal to the next layer. Hidden layer(s) is located in the middle between the input and output layers. Hidden layer contains computational neurons and it hides its desired output. Finally, output layer contains computational neurons and its function to set the output pattern of the whole network. Figure 2.1 shows a FFBP with two hidden layers (Negnevitsky, 2005).

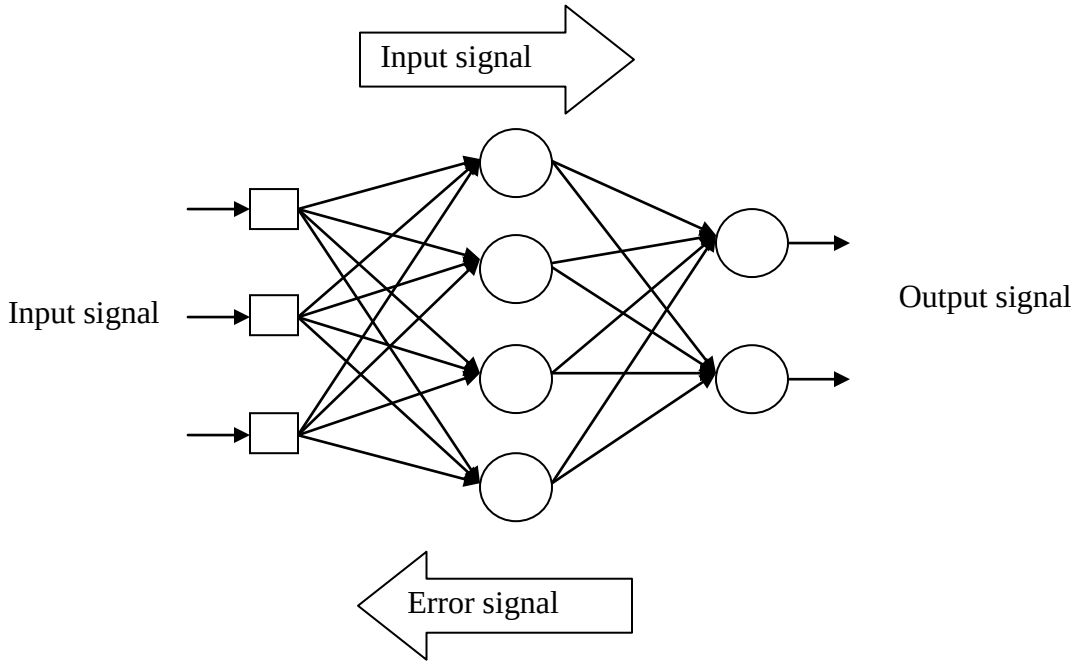

Figure 2.1: Feed-forward Back Propagation Neural Network with two hidden layers.

The number of the neurons in the input layer is determined by the number of data groups. The desired output (target) controls the number of the neurons in the output layer. On the other hand, the determination of the number of hidden neurons is not that straight forward. Normally, trial and error technique is used to investigate the best number of hidden neurons to reach problem optimization without being trapped in local minima.

Back propagation network, like other networks, is specified by the network's architecture, the activation function used by the neurons, and the learning algorithm that sets the procedure for adjusting weights and biases. Usually, sigmoid function is used as the activation function for back propagation networks as follows(Negnevitsky, 2005):

$$Y^{sigmoid} = \frac{1}{1+e^{-x}} , \qquad (2.1)$$

where, *x* is the net input of a neuron, and Y is the output of that neuron.

The learning procedure in FFBP network is divided into two phases. In the first phase, input signals are propagated forward through the network on a layer-by-layer basis until the output pattern is created by the output layer. Then, the error (*e*) between the desired and actual outputs is calculated. This error is back propagated in the second phase to update the weights and biases of each neuron using the delta rule as follows:

$$\Delta w(p) = \alpha \times y(p) \times \partial(p), \tag{2.2}$$

where α is the learning rate, y is the neuron's output at iteration p and $\partial(p)$ is error gradient. To calculate the error gradient of the output neurons, the following formula is used:

$$\partial(p) = y(p) \times [1- y(p)] \times e(p), \tag{2.3}$$

while the error gradient of the hidden neurons is calculated in terms of the back propagated error gradient of the output neurons as follows:

$$\partial(p) = y(p) \times [1- y(p)] \times \sum_{k=1}^{l} \partial_k(p) \times w_k(p), \tag{2.4}$$

where l is the number of neurons in the output layer(Negnevitsky, 2005).

Back propagation network was used for comparison in this research because it is well established method for problem optimization. Furthermore, back propagation networks are clever in recognizing unknown, noisy pattern in short time. Add to that, studies in the literature have been confirming the capability of back propagation networks in speech recognition problems (Love & Kinsner, 1991).

Even though the FFBP network is widely used in many applications, it bears some deficiencies. First, it needs many data (examples) in order to train and test the network. Furthermore, training process is rather slow, but upon reaching to a final solution, network simulation is a matter of seconds. Moreover, the internal mapping actions are not well realized; hence, the convergence of the network to a minimal accepted error is not assured. Also, FFBP technique is prone to being trapped in local minima resulting in ungeneralizable solution (M. K. Alsmadi, Bin Omar, Noah, & Almarashdah, 2009).

To be conclude, back propagation network has few limitations, but it has been proved its capability to deal with many classification problems; reaching to optimum solutions that can be generalized to any other set of data of the same problem.

2.4.2 Elman Network

Elman network, proposed by Elman in 1990, is a type of **R**ecurrent **N**eural **N**etworks (RNN). Also, Elman network is considered as a special type of feed-forward NNs with supplementary memory neurons and local feedback (Yuan-Chu, Wei-Min, & Jie, 2008).

Elman architecture is consisted of four layers: input, hidden, context and output layers. While, the only task of the input layer is signal transmitting from the external environment to the first hidden layer, the output layer function as linear weighted. On the other hand, the hidden layer takes the main task by mapping the input and output layers. The role of the context layer is to memorize the previous output of hidden layer. Obviously, There are adjustable weights and biases linking the adjacent layers (Liang, 2010).

Figure 2.2 depicts the architecture of Elman NN consisting. It is shown that the context layer is entirely connected with all the hidden neurons; such that there is a weight between each context neuron and each hidden neuron. Moreover, it is shown in figure 2.2 that there are recurrent connections from the hidden neurons back to the context neurons (Yuan-Chu, Wei-Min, & Wei-You, 2002).

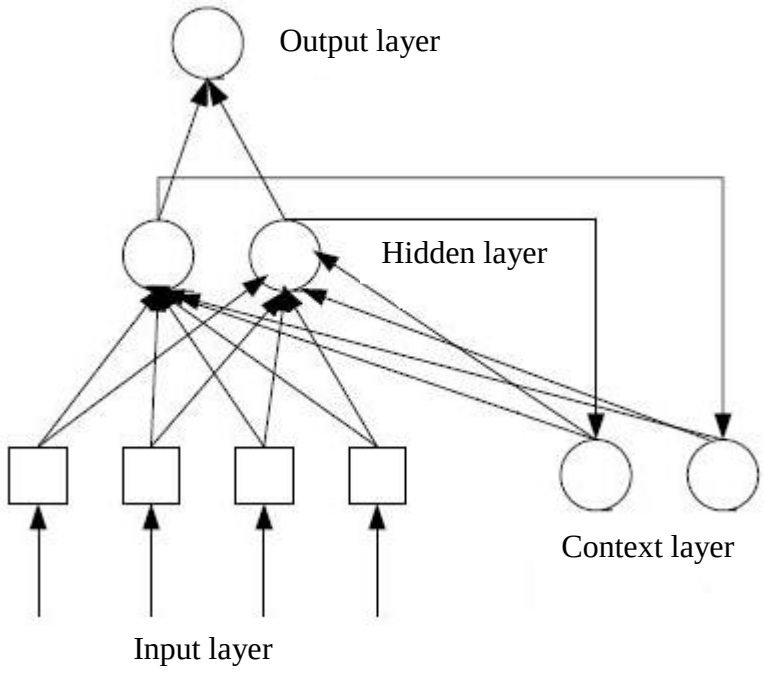

Figure 2.2: The architecture of Elman NN consisting of four layers: input, hidden, context and output.

The weights of the recurrent connections are fixed but the forward weights are updated and trained by using back-propagation. Like FFBP, the learning procedure in Elman network is divided into two phases. In the first phase, the context neurons act as input neurons. The process of calculating the values of each hidden and output neurons is same what is done in FFBP (explained in the previous section). In the second phase, the output of the hidden neurons get transferred to the corresponding context neurons through the recurrent connections. These values are initialized randomly at the first time. Target values of the outputs are used throughout the backward stage of the training, and the forward weights are adjusted by back-propagation (Yuan-Chu, et al., 2002).

Studies in the literature have been confirming that Elman neural network structure has improved training and performance of the network. The most popular used transfer function in hidden layer is the hyperbolic tangent sigmoid transfer function. On the other hand, linear transfer function is normally used in the output layer (Aussem, 1999).

2.5 *K*-Fold Cross Validation

Cross-Validation is a statistical technique for estimating and examining learning algorithms. This is done by dividing the data randomly into two categories: one is used for training a model while the second is used to validate or test the model. The basic type of cross-validation is *k*-fold cross-validation (Krogh & Vedelsby, 1995).

The data in *k*-fold cross-validation is divided randomly into *k* equally sized parts (folds). Then, *k* number of iterations of training and validation are carried out. A different fold of the data is kept for validation in each iteration while the remaining *k*-1 folds are used in the learning process. Finally, the average of cross validation accuracy is calculated.

An example with k = 3 is shown in figure 2.3. The dark parts of the data are used for training while the white parts are used for validation (Kohavi, 1995).

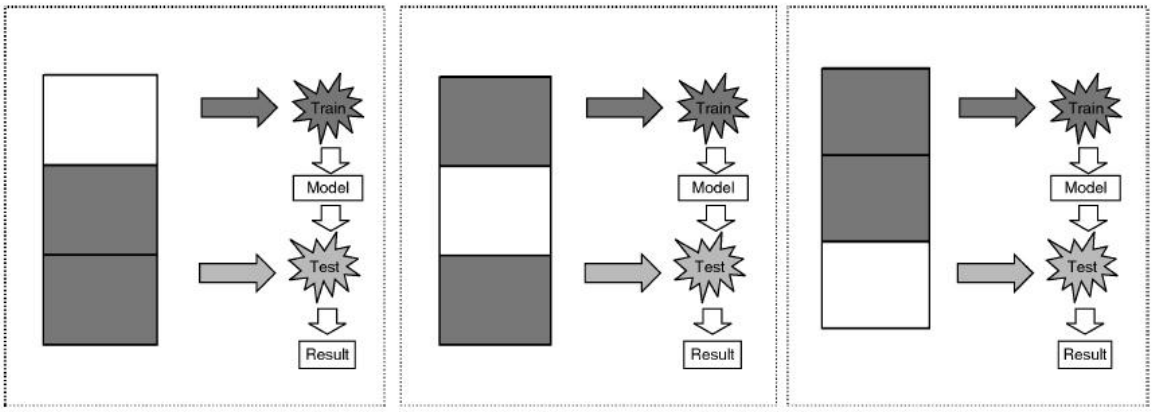

Figure 2.3: Schematic diagram shown the process of 3-fold cross validation.

There are two main goals of cross validation technique:

- To evaluate generalizability of an algorithm.
- To compare the performance of two or more different algorithms and conclude the best algorithm for the available data.

The major drawback of *k*-fold cross validation is the need to an additional computations; because the training process must be performed *k*-times (Faisal, Taib, & Ibrahim, 2010). Another disadvantage is the need to a large amount of data to satisfy the technique.

Although, 10-fold and 5-fold are the widely used cross validation methods, many studies found in the literature assured that there is no significant statistical difference between the use of 10-fold, 5-fold or 3-fold cross validation techniques (Feng et al., 2008).

Therefore, the current study implemented 3-fold cross validation in which 2/3 of the data (240 examples) was used for training and the rest 1/3 of the data (120 examples) was used for validating.