Indonesian Speech Recognition System Using Discriminant Feature Extraction – Neural Predictive Coding (DFE-NPC) and Probabilistic Neural Network

Untari N. Wisesty, Thee Houw Liong, Adiwijaya
Telkom Institute of Technology
Bandung, Indonesia
untarimail@gmail.com, adiwijaya007@yahoo.co.id, thehl007@gmail.com

***Abstract*— Along with advances in information technology, it has been developed the technology to facilitate human life, one of which is speech recognition. Speech recognition is widely applied to speech to text, speech to emotion, in order to make gadget and computer easier to use, or to help people with hearing disability. However, the development of speech recognition to produce the text from the input voice has not well developed because of time processing. This is certainly going to make the animators and engineers need more time using speech recognition. Therefore, a method is needed to solve the time processing problem and with a good accuracy.**

This study proposes a speech recognition system using Discriminant Feature Extraction – Neural Predictive Coding (DFE-NPC) as feature extraction and Probabilistic Neural Network as recognition method. This system can accelerate time processing because it only uses one iteration in training process. Time processing of proposed method is decreased significantly until 1:95 compared to Fuzzy Hidden Markov Model. The best accuracy of the system is 100% when number of class is 2 and 3, and the worst one is 56% when number of class is 10.

Keywords- Speech Recognition System; DFE-NPC; PNN; time processing;

I. INTRODUCTION

Along with advances in information technology, it has been developed the technology to facilitate human life, one of which is speech recognition. Speech recognition is widely applied to speech to text, speech to emotion, in order to make gadget and computer easier to use, or to help people with hearing disability. However, the development of speech recognition to produce the text from the input voice has not well developed because of time processing. This is certainly going to make the animators and engineers need more time using speech recognition. Therefore, a method is needed to solve the time processing problem and with a good accuracy.

Speech recognition system has been developed using the Hidden Markov Model [3]. Hidden Markov methods are easy to apply and have a training algorithm to estimate model parameters for the set of voice data. These methods have flexible architectures in both size and type which are appropriate to the type of words and sounds. The speech recognition system has also been developed with a stereo vision neural network model [4]. These models apply the 3-layered neural net (3LNN) and equation stereo vision to obtain better recognition results than the hidden Markov methods.

Based on several methods described above, there are still some drawbacks, i.e. speech recognition systems are still focused on the problems of accuracy and less concerned about the speed of the system's processing. Therefore, this study tries to develop speech recognition systems with good accuracy and minimum time processing. This study proposes a speech recognition system using Discriminant Feature Extraction – Neural Predictive Coding (DFE-NPC) as feature extraction and Probabilistic Neural Network as recognition method. This system can accelerate time processing because it only uses one iteration in training process.

II. ALGORITHM

In this research uses Discriminant Feature Extraction – Neural Predictive Coding (DFE-NPC) as feature extraction and Probabilistic Neural Network as recognition method.

A. DFE-NPC

DFE-NPC is one of speech signal feature extraction based on Linear Predictive Coding (LPC). Here is the algorithm of DFE-NPC:

1. Pre-emphasis :

This process is used to equalize spectral signal and to omit top values in the spectral signal, so it will be easier to decide boundary of signal for next process.

$$\tilde{s}(n) = s(n) - \tilde{a}\, s(n-1) \qquad (2.1)$$

where:

$s(n)$ is n^{th} sample

$\tilde{a}$ is adjust parameter (default = 0,95)

2. Frame blocking :

Frame blocking divides the result of pre-emphasis $\tilde{s}(n)$ into frames. Each frame has N sample and separate about M sample. If M<<N then the spectral prediction of frame by frame is better.

3. Windowing :

This process minimizeS discontinuity between start point and end point of each frame, by Hamming Window equation:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), 0 \le n \le N-1 \qquad (2.2)$$

$$\tilde{x}(n) = s(n).w(n) \qquad (2.3)$$

where:

w(n) is hamming window

s(n) is sample frame

4. DFE-NPC phase :

✓ Parameterization phase

In this phase, all weight in first layer is trained using prediction error minimization criteria.

$$Q = \sum_{k=1}^{K}(y_k - \hat{y}_k)^2 = \sum(y_k - F(y_k))^2 \qquad (2.4)$$

where:

$F_{w,a}(y_k) = Ha \circ Gw(y_k)$ (2.5)

$\tilde{y}_k = Ha(z_k)$ (2.6)

$z_k = Gw(z_k)$ (2.7)

$y_k = [y_{k-1}, y_{k-2}, \dots, y_{k-\lambda}]^T$ (2.8)

where:

λ : length of windows prediction

w : hidden layer weight

a : output layer weight

σ : activation function

Coding vector: output layer weight (a)

✓ Coding phase

In this phase, weight of output layer is calculated in a form of vector code and prediction of error minimization criteria, but only update the weight of output layer by Levinson algorithm. The initial output weight is random, w has been known, and a is calculated by equation NPC and LPC model.

$\Theta = w \bullet a$ (2.9)

$a = w^{-1} \bullet \Theta$ (2.10)

where: Θ : LPC parameter

This is the architecture of DFE-NPC [3]:

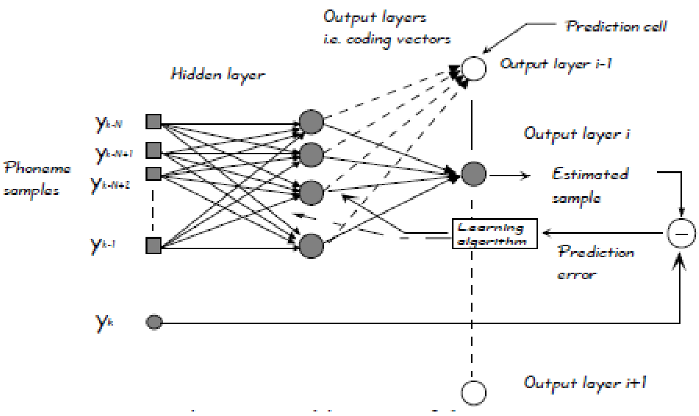

Figure 2.1: Architecture of DFE-NPC

B. Probabilistic Neural Network (PNN)

Probabilistic Neural Network (PNN) is one of artificial neural network model based on probabilistic density function. This classification model has a good performance in accuracy rates and training speed because it only needs one iteration. A smoothing parameter (σ) manages the network which is influenced by every pattern.

Bayes method classifies pattern using a decision rule that minimizes expectable risk. For example, there are n class, C_0, C, C_2,…,C_{n-1}, and observed pattern which is random variable x with m-dimension and conditional density function x. if the pattern is from C_k class, denoted by $p(x|C_k)$. By implementing the first rule of Bayes, then the probability of x in C_k class, denoted by:

$$P(C_k|x) = \frac{p(x|C_k)}{p(x)} \qquad (2.11)$$

where:

p(x) is probability of x

From that case, it can be formulated by the common way to minimize the risk is by minimizing the probability. Bayes decision rule is used to decide class C_k by choosing the highest $Pr(C_k|x)$.

$$d(x) = C_k \ if \ p(x|C_k)\Pr(C_k) > p(x|C_j)\Pr(C_j), for \ all \ j \ne k \qquad (2.12)$$

The probabilistic neural network model made by Cain allows each class to have a smoothing parameter, σ_k, which is different from other and implement learning algorithm to obtain σ_k automatically. If each class has a smoothing parameter, then the probabilistic density function is denoted by:

$$p(x|C_k) = \frac{1}{(2\pi)^{m/2}\sigma_k{}^m|C_k|}\sum_{\rho_i \epsilon C_k}\exp\left[-\frac{||x-w_i||^2}{2\sigma_k{}^2}\right] \qquad (2.13)$$

or:

$$p(x|C_k) = \left(\frac{m}{|C_k|}\right)(\sigma_k{}^m|C_k|)^m(1-(\sigma_k{}^m|C_k|)^{(m-|x-w_i|)} \qquad (2.14)$$

where $|C_k|$ is the number of training pattern in class C_k, m is vector dimension input pattern, and w_i is weight vector in i^{th} training pattern.

The training algorithms that can adjust value of σ_k automatically create the network form parameter for each class in second step of training process. This is the training algorithm of probabilistic neural network.

{First Step}

For each pattern of ρ_i

  $w_i = \rho_i$

  Build pattern unit with input weight vector wi

  Connect pattern unit to summing unit for each class

End

Calculate $|C_k|$ for each summing unit

{Second step}

For each pattern ρ_i
  k = class ρ_i
  Find shortest distance, di, to pattern in class k
  dtot[k] = dtot[k] + d_i
End
For each class k
  σ_k = (g . dtot[k]) / $|C_k|$
  Calculate probabilistic density function
End
Find the highest probability from all class

Smoothing parameter of each class is the multiplication between a constant and the average of minimum distance of training pattern in same class. So, the average of minimum distance between pattern vectors in class C_k is:

$$d_{avg}[k] = \frac{1}{|C_k|} \sum_{\rho_i \in C_k} d_i \qquad (2.21)$$

where d_i is nearest distance pattern ρ_i with other pattern in class C_k. So that, smoothing parameter σ_k for class C_k is:

$$\sigma_k = g \,.\, d_{avg}[k] \qquad (2.22)$$

where g is smoothing parameter and d_{avg} is average distance.

Choosing constant g is used to create high classification accuracy on the network, because constant g is affected by the number of class, training pattern dimension, and number of training set.
The architecture of probabilistic neural network has 4 layers, contained of:

1. Input layer, that contain m unit and receive x input vectors.
2. Pattern unit layer that has full connection to input pattern.
3. Summing result unit layer that has full connection to each class.
4. Decision layer to choose the highest values. [9]

Here is the architecture of probabilistic neural network:

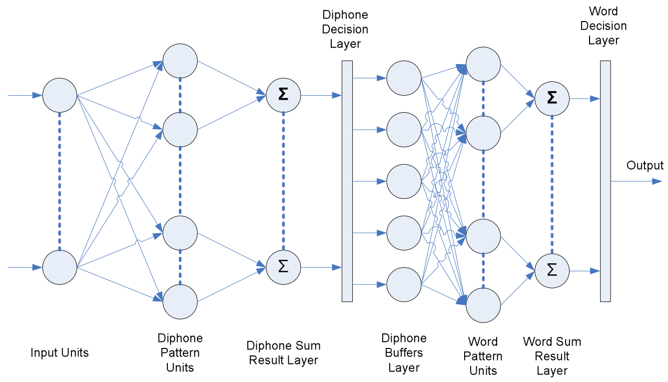

Figure 2.2: Probabilistic Neural Network Architecture

III. EXPERIMENT AND ANALYSIS RESULTS

A. Comparison between proposed method and previous method (Fuzzy Hidden Markov Model)

This experiment is conducted to learn the improvement of proposed method in time processing. By using same data and the most optimal parameter of each method, this experiment compares performance of PNN and FHMM as a speech recognition system. Here is the result of the experiment:

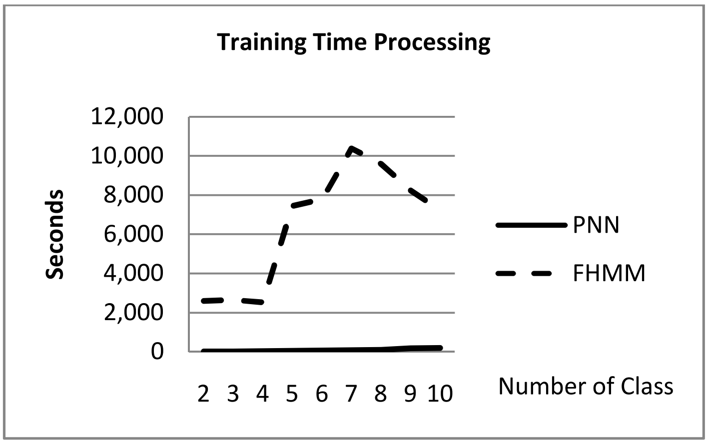

Figure 3.1 : Comparison of time on training process, the axis x denote number of class, axis y denote the time processing (seconds), solid line is result of PNN, and dash line is result of FHMM.

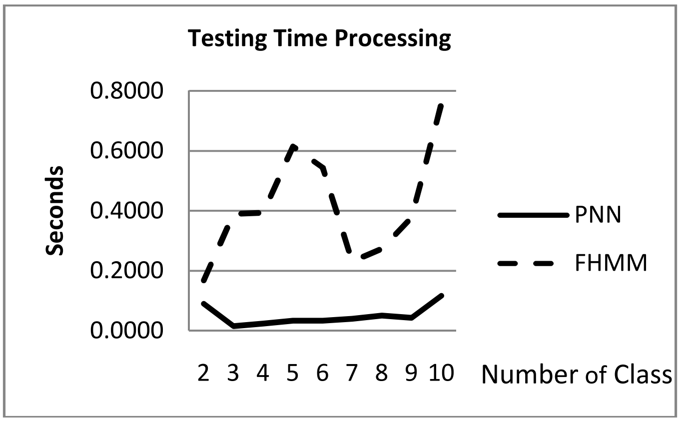

Figure 3.2 : Comparison of time on testing process, the axis x denote number of class, axis y denote the time processing (seconds), solid line is result of PNN, and dash line is result of FHMM.

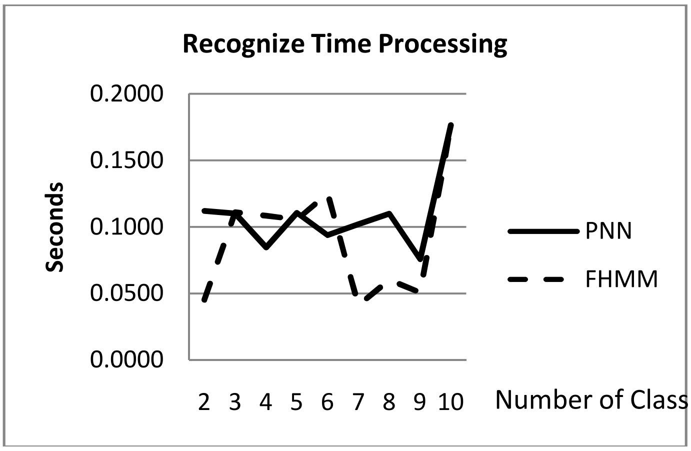

Figure 3.3 : Comparison of time on recognized process, the axis x denote number of class, axis y denote the time processing (seconds), solid line is result of PNN, and dash line is result of FHMM.

Based on the comparison result above, processing time of training and testing of PNN is faster than FHMM. It is because PNN only use one iteration in training process. Whereas, the recognized time processing is almost the same because in recognized process, there are only testing data and calculating the class of data. The time difference of this process is only reach 0.14 seconds and it is influenced by computer activity because at that time the memory of computer is used by other programs, causing the proposed system must wait until the memory is free.

B. Comparison between Normal and Bernoulli Distribution

Table 4-2 Comparison Result between Bernoulli and Normal Distribution

No	Number of Class	Bernoulli		Normal	
		Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy
1	2	1	1	0,966667	0,8
2	3	1	1	0,866667	0,466667
3	4	1	0,9	0,825	0,45
4	5	1	0,92	0,748387	0,6
5	6	1	0,766667	0,691892	0,333333

Based on the table 4.2, the accuracy of Bernoulli distribution is higher than normal distribution. It is because Bernoulli distribution can cover the data that have certain declination.

C. Performance analysis of propose method

In this experiment, there are some variables that used to gain the best performance:

- Number of class: 2, 3, 6, 10.
- Number of cluster: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50
- Smoothing parameter: [0, 1], that increase 0.01 for each observation. So, each variation of number of class and number of cluster will be evaluated about 100 times.

Here are the results of the experiment above:

- The training and testing accuracy :

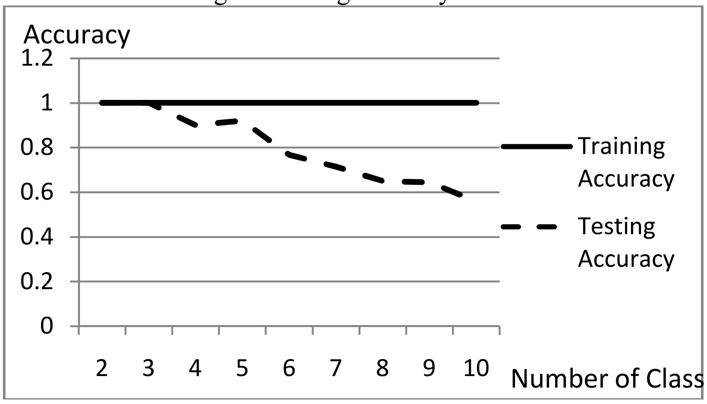

Figure 3.4: Training and Testing Accuracy, the axis x denote number of class, axis y denote the accuracy, solid line is training accuracy, and dash line is testing accuracy.

From the result above, testing accuracy decreases if the number of class increases. It is happened because of the increasing variation of the data, so that the smoothing parameter in PNN increases as the number and class and they are difficult to be classified.

- The time processing:

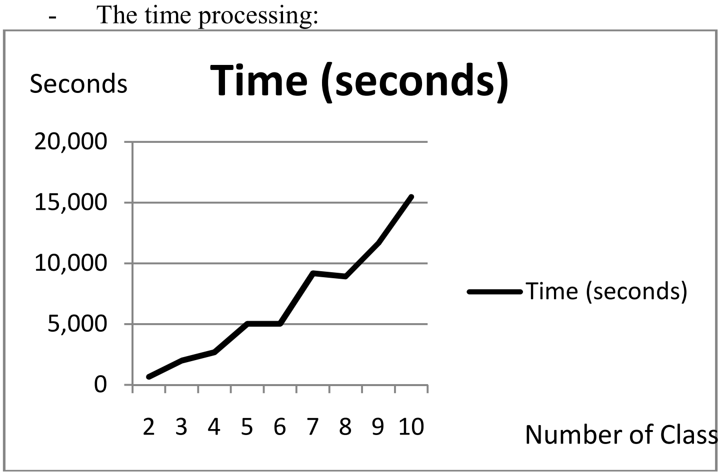

Figure 3.5: Time Processing for the Experiment, the axis x denote number of class, axis y denote the time processing (seconds), and black line is result of PNN.

Time processing is affected by the number of data used in training and testing data that is increased linearly. Time processing above is in training and testing process when finding the most optimal parameter. The vector quantization is the most affecting factor on the time processing.

- The number of cluster is shown in figure 4.7

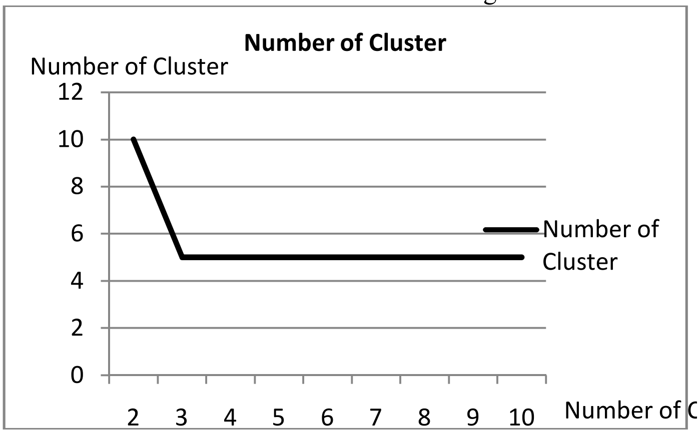

Figure 3.6: Result of Number of Cluster Experiment, the axis x denote number of class, axis y denote the number of cluster, and black line is result of PNN.

Number of cluster is used in vector quantization to build the codebook. In this study the most optimal number of cluster is obtained at 10 for 2 classes and 5 for 3 classes until 10 classes. It shows that the probabilistic neural network only needs small variance of input to classify the data.

- The Smoothing Parameter :

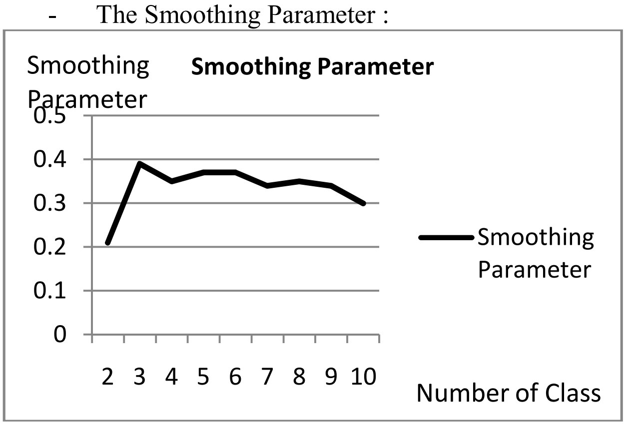

Figure 3.7: Result of Smoothing Experiment, the axis x denote number of class, axis y denote the smoothing parameter, and black line is result of PNN

Smoothing parameter is a parameter that affects classification capability of probabilistic neural network. Based on the graph above, the most optimal smoothing parameter is obtained about 0.21 until 0.39. It shows that in this study, the higher the smoothing parameter, the rougher the classification and the lower the accuracy will be acquired.

IV. SUMMARY

Based on the experiment results and analysis, the thesis can be summarized into several conclusions:

1. Time processing of proposed method is decrease significantly until 1:95 compared to Fuzzy Hidden Markov Model.
2. If number of class increase, then the accuracy of system decreases.
3. The time processing is linearly increased in proportion to the number of data.
4. The accuracy of system is influenced by number of cluster and smoothing parameter. In this study, the most optimal number of cluster is obtained at 5 and 10, and smoothing parameter is obtained at 0.21 until 3.9. The smoothing parameter also influences the probability density function for each class.
5. The best accuracy of the system is 100% when number of class is 2 and 3, and the worst one is 56% when number of class is 10.

REFERENCES

[1] B.H. Juang and Lawrence R. Rabiner: Automatic Speech Recognition – A Brief History of The Technology Development. Georgia Institute of Technology and Rutgers University and The University Of California, 2004.

[2] Breebaart, Jeroen, Martin McKinney, “Fearures for Audio Classification”, Philips Research Laboratories.

[3] Chetouani, M., B. Gas, J.L. Zarader, "Maximization of the Modelization Error Ratio for Neural Predictive Coding", Universite Paris VI, 2003.

[4] Chetouani, M., B. Gas, J.L. Zarader, "Learning Vector Quantization and Neural Predictive Coding for Nonlinear Speech Feature Extraction", Universite Paris VI, 2003.

[5] Kocsor, Andras, Gabor Gosztolya, “The use of Speed-up Techniques for a speech Recognizer System”, Springer Science+Bussiness Media, LLC 2008.

[6] Rabiner, Lawrence, Biing-Hwang Juang, “Fundamentals of Speech Recognition,” Prentice-Hall International, 1993.

[7] Rabiner, Lawrence R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Procedings Of The IEEE,, Vol. 77, No.2, February 1989.

[8] Stephen E. Levinson, Lawrence R. Rabiner, Aaron E. Rosenberg, and Jay G. Wilpon: Interactive Clustering Techniques for Selecting Speaker-Independent Reference Templates For Isolated Word Recognition. IEEETransactions on Acoustics, Speech, and Signal Processing, Vol. Assp-27, 1979.

[9] Suyanto, “Artificial Intelligence, Searching, Reasonning, Planning, and Learning”, Informatika Bandung, 2007.

[10] Yulita, Intan Nurma, “Fuzzy Hidden Markov Models for Indonesian Speech Classification, Journal of Advance Computational Informatics and Intelegent Informatics, vol 16 no.3, 2012.