2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Improved Keyword Spotting System by Optimizing Posterior Confidence Measure Vector Using Feed-forward Neural Network

Yuchen Liu, Mingxing Xu and Lianhong Cai

Abstract— **In this paper, a novel method based on feed-forward neural network is proposed to optimize the confidence measure for improving a mandarine keyword spotting system. Keyword spotting is to detect the occurrences of a pre-defined list of keywords in the input speech, and confidence measure is an critical part in the verification stage of keyword spotting. Posterior confidence has been widely used and was verified to be effective. In some previous works, the optimization of posterior confidence has been proposed, which linearly transforms the phone-level confidence into the word-level confidence. On this basis, we propose a neural network based method that make a non-linear transformation. In addition, a sparse activation and back-propagation strategy is proposed to make this method feasible and work fast. In the experiments, the proposed method is compared to other two previous methods. To evaluate performance, two most commonly used measures are considered: AUC and EER. The experimental result shows that the proposed method is effective and achieved the best performance among three methods.**

I. Introduction

KEYWORD Spotting System(KWS) is used to detect the occurrences of pre-defined keywords in continuous speech utterance. This technology has been widely used in many application areas such as speech command control, voice message classification, audio information retrieval and automatic queries system.

There are three major categories of keyword spotting system[1]: LVCSR-based, phone-lattice based and acoustic keyword spotting. In the experiments of [1], they investigate that the phone-lattice based method has the worst performance and the best performance was achieved by the LVCSR-based method. There are two stages in the LVCSR-based method. First, the system proceed a large vocabulary continuous speech recognition on the speech utterance, and then grep the keywords in hypothesized results which can be 1-best answer, n-best hypothesized lists or word lattice. In the task of keyword spotting, it's unnecessary to recognize the whole sentence of the utterance. Speed and quick response of the system is supposed to be more important in many applications of embedded system, e.g., set top box and unconnected pad.

The performance of acoustic keyword spotting system approach closely to LVCSR-based method, and could be running far more quickly in realtime because of the no considerations on a large language model, so that it's chosen as our system. A variety of methods have been proposed to improve acoustic keyword spotting. In [2], confusion garbage model was developed to absorb similar pronunciation words confused with the specific keywords of a task. The combination of acoustic and LVCSR based keyword system was proposed as a method that the lattice generated from LVCSR was used to improve performance[3]. Even there are so many methods, the most widely used is the calculation and optimization of confidence measure.

Confidence measure plays an important role in the verification stage of keyword spotting. In the hypothesized detections from the decoder of acoustic keyword spotting, some are correct hits and others are false alarms. As usually know, the optimal target of the confidence measure is to give higher confidence measure to the correct hits and lower confidence measure to the false alarms. Based on this assumption, a threshold T is usually set that a hypothesized detection is supposed to be accepted if it's higher than T, and on the contrary condition that it is lower than T, it should be rejected. The calculation of a reasonable confidence measure has a great progress in previous works.

In early years, LR-based confidence measure was proposed. LR means likelihood-ratio which represents the ratio of keyword's likelihood to the likelihood of non-keyword. The modelling of non-keyword is the major problem in LR-based method. Some modelling methods were proposed such as online dynamic filler model[4] and anti-subword model[5]. In later years, posterior probability based confidence measure overcome the disadvantage of requiring alternative model in LR-based methods, and in the meanwhile, posterior confidence achieve a great much better performance than the LR-based methods[6].

There are three levels in the calculation of posterior probability based confidence measure. Frame-level logarithm posterior probability is calculated using acoustic model with a strategy called catch-all model. Phone-level confidence is estimated from the frame-level confidence. Although the average of phone-level confidence is natural for the establishment of word-level confidence, some optimized methods have been proposed for this step of combination. Classification of three distinct average values using support vector machine has been proposed in [7]. Two weighted average based methods have been proposed which have different objective functions called MCE[8] and AUC[9]. Essentially, these weighted average methods are linear regression based methods that is linearly mapping the phone-level confidence

Yuchen Liu, Mingxing Xu and Lianhong Cai are with Key Laboratory of Pervasive Computing, Ministry of Education Tsinghua National Laboratory for Information Science and Technology(TNList), Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: liuyuchen921@163.com).

This work was partially supported by the National Natural Science Foundation of China (No.61171116) and National Basic Research Program of China (973 Program, No.2012CB316401).

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

to the word-level confidence.

In this paper, we propose using feed-forward neural network for the optimization of the phone-level confidence measure into the word-level. The phone-level confidence vector is constructed for the linear regression and the neural network. They would map a confidence vector to its confidence measure. For fast training and efficient computation, a strategy called sparse activation and sparse back-propagation is proposed.

The rest of this paper is organized as following: Section 2 introduces the Keyword Spotting System; Section 3 explains how to calculate the posterior probability based confidence measure and the weighted mean based methods. Section 4 proposes the feed-forward neural network based method, including the trick of sparse activation and back-propagation. Section 5 illustrates the experimental results. Conclusion is draw in the last Section 6.

II. Keyword Spotting System

The architecture of the keyword spotting system is shown in Fig. 1. In the front-end, 12-dimensional MFCC, 1-logarithm energy and 1-pitch with their first and second derivatives form 42-dimensional feature vector which is extracted by the feature extraction stage. The silence detection will mark every silence frame and these marks will be referred in the decoder. The frame length is 24ms and the frame shift is 12ms.

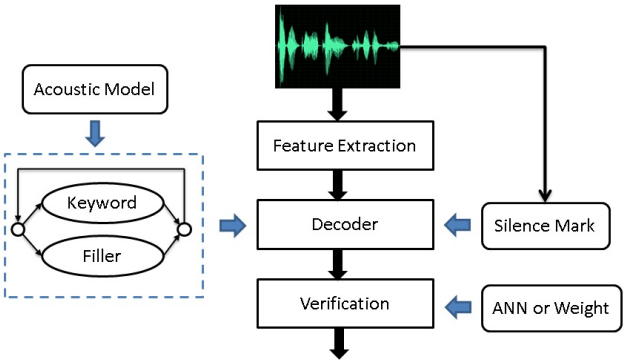

Fig. 1. The architecture of the keyword spotting system

For the decoder, a keyword-filler network is supposed to be constructed from the acoustic model which is tied state tri-phone modelled HMM/GMM. The keyword part in the network is the parallel connection of all keywords' HMM. The filler part is the parallel connection of all syllables' HMM. Keywords' and syllables' HMM are all built from the series connection of their tri-phones' HMM. The decoder stage conducts a cross-word search on the network while processing frames. After that, some hypothesized detections would be sent into the last verification stage. The final results are the hypothesized detections accepted by the verification stage. In the next section, we will discuss the principle and method of calculation confidence in the verification stage.

III. Posterior Confidence Measure

Confidence measure is the critical part of the verification stage and affect the global performance of the keyword spotting system on the foundation of the decoder. Posterior confidence can be seen as calculation at three levels: frame, phone, word. We will introduce this process from bottom to top.

A. Frame-level Posterior Probability

At each frame t, the frame-level posterior probability of a state s to the observation o_t can be defined as:

$$p(s|o_t) = \frac{p(o_t|s)p(s)}{p(o_t)} \tag{1}$$

where $p(o_t|s)$ is the likelihood of the observation o_t with respect to the state s, $p(s)$ is prior probability of the state s, which we assume to be all equal, and $p(o_t)$ is calculated using a catch-all model[6][10][11] as:

$$p(o_t) = \sum_{i=1}^{N_s} p(o_t|s_i) \tag{2}$$

where N_s is the number of all states and s_i is the i-th state. In [8], they proposed an alternative method that is more effective. It convert the equation (2) to:

$$p(o_t) = \sum_{i=1}^{N_s} p(o_t|s_i) I_A(s_i) \tag{3}$$

where $I_A(s_i) = 1$ only if s_i is active in the processing at the frame t while decoding and $I_A(s_i) = 0$ if s_i is not active, i.e, to accumulate of only the active states. This estimation seems less complete than the accumulation of all states, but the result in [8] testify it to be more effective, may be cause of excluding many odd states by the beam pruning in the decoder.

B. Phone-level Confidence Measure

To compute the word-level confidence, the phone-level confidence is supposed to be calculated at first. Assume a hypothesized keyword W is composed of N_W phones, and the i-th corresponding tri-phone is tph_i^W. The phone-level confidence of tph_i^W is the duration mean of the frame-level logarithm posterior probabilities:

$$CM(tph_i^W) = \frac{1}{t_i^e - t_i^s + 1} \sum_{t=t_i^s}^{t_i^e} \log p(s_t|o_t) \tag{4}$$

where t_i^s and t_i^e are the start and end frame of the i-th tri-phone tph_i^W respectively, o_t is the observation at the frame t, and s_t is the state aligned at the frame t according to a Viterbi re-alignment.

C. Word-level Confidence Measure

The confidence measure for a hypothesized keyword W is combined from its phone-level confidences. The baseline used in this paper is the average of the phone-level confidence:

$$CM(W) = \frac{1}{N_W}\sum_{i=1}^{N_W} CM(tph_i^W) \tag{5}$$

It has been found that phone-level confidence should contribute to word-level confidence in different degrees. Thence, a weighted average of phone-level confidence is employed to acquire word-level confidence:

$$CM(W) = \frac{1}{N_W}\sum_{i=1}^{N_W} (a_{tph_i^W} CM(tph_i^W) + b_{tph_i^W}) \tag{6}$$

The weights $a_{tph_i^W}$ and the bias $b_{tph_i^W}$ is supposed to be optimized by the gradient descent of an objective function as an optimal target. Two such objective functions have been proposed. The MCE function can be expressed as[8]:

$$d(W) = \sigma((CM(W) - C) \times Sign(W)) \tag{7}$$

where $\sigma(\cdot)$ is the sigmoid function, C is a threshold which will be varying in the gradient descent, and $Sign(W) = 1$ if W is incorrect, else $Sign(W) = -1$. The optimal target is to minimize the MCE function $d(W)$. It's a disadvantage to maintain this varying threshold, whose initial value is unpredictable for wandering wildly in the iterations of the gradient descent.

The AUC objective function is to maximize the area under the ROC curve directly, which is the metric of the performance. This function can be defined as[9]:

$$A = \frac{\Theta_{max}}{|H^+| \cdot |H^-|} \sum_{u \in H^+} \sum_{v \in H^-} \sigma(CM(u) - CM(v)) \tag{8}$$

where, Θ_{max} is the maximal hit rate which the keyword spotting system could achieve with no considerations on false alarms, H^+ is the set of all positive hits and H^- is the set of all false alarms. The weakness of this method can be seen in the objective function, that is calculated from each pair of the positive hits to the false alarms. Therefore, the computational complexity of training n samples will be $O(n^2)$, that it can not be applied to a large account of samples.

The common essential weakness of these two weighted mean based methods is that they are all linear transformations from the phone-level confidence to the word-level. To compete this disadvantage, we employ a feed-forward neural network based method to implement the nonlinear transformation from the phone-level confidence to the word-level.

IV. Optimization using Feed-Forward Neural Network

A. Phone-Level Confidence Vector

In the weighted average methods, there is a weight and bias for each tri-phone. It can also be seen as a weight vector and a bias vector whose dimension is corresponding to the tri-phone. Assume that a total of L tri-phones occurred in the acoustic model and they are indexed in the range of 1 to L in some order. Hereupon, the phone-level confidences of a keyword W can be built into a vector V^W which has a form like:

$$V^W_{I(tph_i^W)} = CM(tph_i^W)/N_W \tag{9}$$

where tph_i^W is the i-th tri-phone of W and $I(tph_i^W)$ is its index. It's reasonable to set $V_j^W = 0$ if the index j is not covered by the tri-phones in W. It's necessary to notice that the phone-level confidence vectors V^W are all extremely sparse, cause of $N_W \ll L$ all the time. Furthermore, it will be shown later that this property of sparsity can be fully used to speed up the calculation and training with the neural network.

For a keyword W, we send its phone-level confidence vector as sample into the input layer of a feed-forward neural network and at the output layer, get the activations of the two neurons , which are denoted as o_0 and o_1 respectively. The target of o_0 and o_1 in the training will be set as $t_0 = 1.0$ and $t_1 = 0.0$ if it's a positive hit, and $t_0 = 0.0$ and $t_1 = 1.0$ on the contrary. Therefore, it's natural to use $o_0 - o_1$ as the word-level confidence. A demonstration of this transformation is illustrated in Fig. 2.

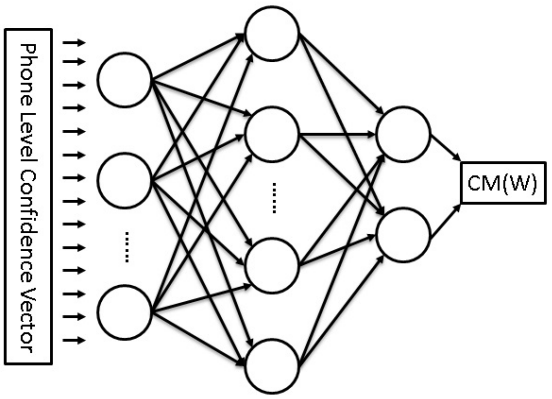

Fig. 2. Transformation using the neural network

B. Sparse Activation and Back-propagation

The major cost of the feed-forward neural network's activation and back-propagation in this work is between the input layer and the first hidden layer, since the input vector is very long. To accelerate the activations, we first analyze the expression of a neuron's activation in the hidden layer:

$$o(j) = \sigma(\sum_{i=1}^{L} w_{ji} V_i^W) \tag{10}$$

where i is a neuron in the input layer, j is a neuron in the first hidden layer, and $o(j)$ is the output activation of the neuron j. In fact, the sample vector V_i^W is very long but sparse, i.e, except the N_W tri-phones' index position, the value in the

vector is filled with zero. Using this property, we can refine the expression:

$$o(j) = \sigma(\sum_{p=1}^{N_W} w_{jI(tph_p^W)} V_{I(tph_p^W)}^W) \tag{11}$$

For training a feed-forward neural network, the so-called back-propagation is always applied. The essential of back-propagation is the gradient descent to the objective function defined at the output layer. The objective function used in this work is the mean-square error:

$$E = \frac{1}{2} \sum_{k=1}^{2} (t_k - o_k)^2 \tag{12}$$

By the partial differential of the objective function, we can see the increment of a weight from input to the hidden layer:

$$\Delta w_{ji} = \frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial o(j)} \frac{\partial o(j)}{\partial w_{ji}} \tag{13}$$

$$\frac{\partial o(j)}{\partial w_{ji}} = \gamma o_j (1 - o_j) V_i^W \tag{14}$$

where E is the objective function, γ is the smoothing parameter in the sigmoid function. Neglected of the first derivative item of the increment, we discover that the variation of the weight is relevant to V^W. Hence $\Delta w_{ji} = 0$ if $V_i^W = 0$, that the number of weights needed to be updated processing one sample is $N_W H$, where H is the size of the hidden layer. Finally, the sparse activation and back-propagation reduce the computational complexity between the input and hidden layer to $O(N_W H)$, which is far more quickly than the naive one, since $N_W \ll L$.

V. Experiments

The samples for training the neural network are generated by running keyword spotting on a mandarine speech set which comes from two databases. One is a telephone mandarine speech database called TeleDB, and another is a labelled reading-style mandarine speech database called 863DB. There are totally about 43 hours speech, consisting of 35845 utterances pronounced by 159 speakers.

In order to generate enough samples to cover all tri-phones, keyword spotting system is running on the speech set many times with different keyword lists which are random picked up. At last, after about 500 passes, $2,181,729$ samples are generated, in which there are $225,099$ positive hits and $1,956,630$ false alarms. To avoid over-fitting, another evaluation set is built to evaluate the performance in the training process. The length of the evaluation set is 18 hours, which consist of 14623 utterances pronounced by 80 speakers. The keyword spotting system is running on the evaluation speech set 2 times with random chosen keyword lists.

A distribution of the occurrence number of each tri-phone in the training samples is demonstrated in Fig. 3. The number of all tri-phones in the model is 19480. Occur time of a tri-phone is ranged under 10000 times and most commonly under 2000 times. A histogram of the number of tri-phones with different occur times is illustrated in Fig. 4. There are less tri-phones with higher occur times and the distribution of this histogram appears as a long tail.

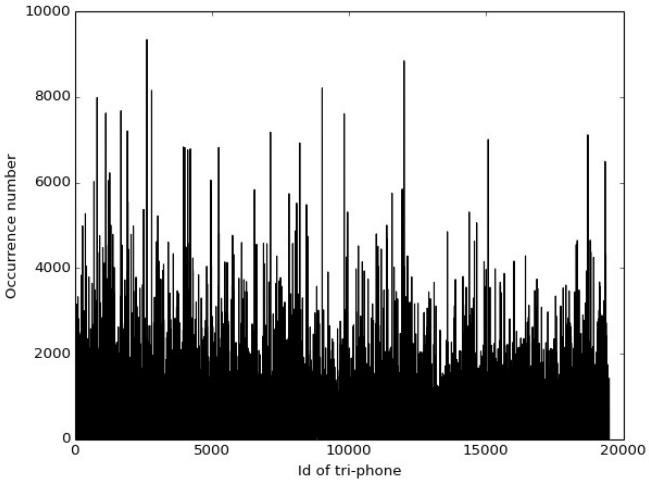

Fig. 3. Occurrence number of each tri-phone

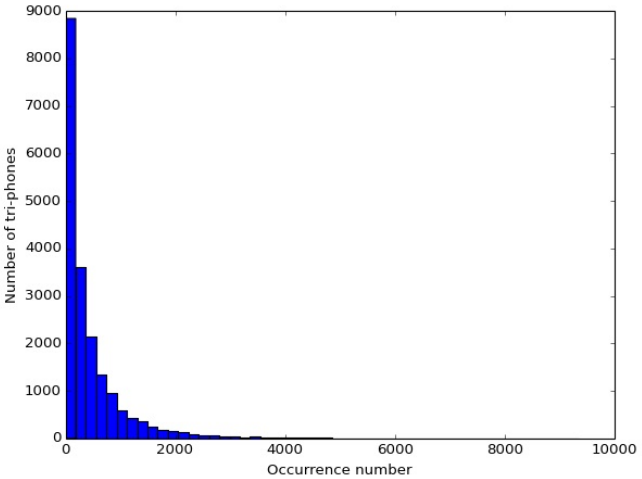

Fig. 4. Number of tri-phones within different occurrence numbers

The test set used in this work is a 4-hours speech set, which consist of 3190 utterances pronounced by 31 speakers. There are 4 keyword lists random chosen for testing where each list has 100 keywords.

For the evaluation of the performance, some measures have been used in different works. In this work, we use two measures which are most commonly used: AUC and EER. In early years, Figure of Merit(FOM) is commonly used in measuring the performance of keyword spotting. FOM is the average hit rate at 10 false alarm rate:$1, 2, ..., 10$. The hit rate is the number of correct detections divided by the number of all occurrences of keywords in speech set. The false alarm rate is the number of false alarms divided by the speech hours and the number of keywords. The ROC

curve indicates the relationship of the hit rate with respect to the false alarms rate. The area under the ROC curve is called AUC. It's obvious to see that FOM is an approximation of AUC. In another aspect, EER is the equal error rate on the DET curve. The DET curve indicates the relationship between the false acceptance rate(FAR) and false rejection rate(FRR). The FAR is the number of total false acceptances divided by the number of total false attempts, and FRR is the number of total false rejections divided by the number of total true attempts. The EER is the error rate at the point of DET where FAR=FRR.

The first experiment expects to verify that the neural network based method is effective and the performance is increasing with respect to the decreasing of the mean-square error rate of the neural network. Fig. 5 shows the variation of AUC, EER and mean-square error rate in the process of training iterations on the evaluation set. It can be seen that the mean-square error rate decrease, AUC increase, and EER decrease while the iteration proceeding. This trend deduces that according to whatever measures, the training process of the neural network enhance the performance accompanied while the objective error descending. Therefore, it's also to be noticed that after about 50 iterations, the AUC, EER and mean-saure error tend to wave at some level, which indicates the fitting of the model with the data. Therefore, the training iteration will be stopped after 50 iterations or the AUC and EER are becoming bad.

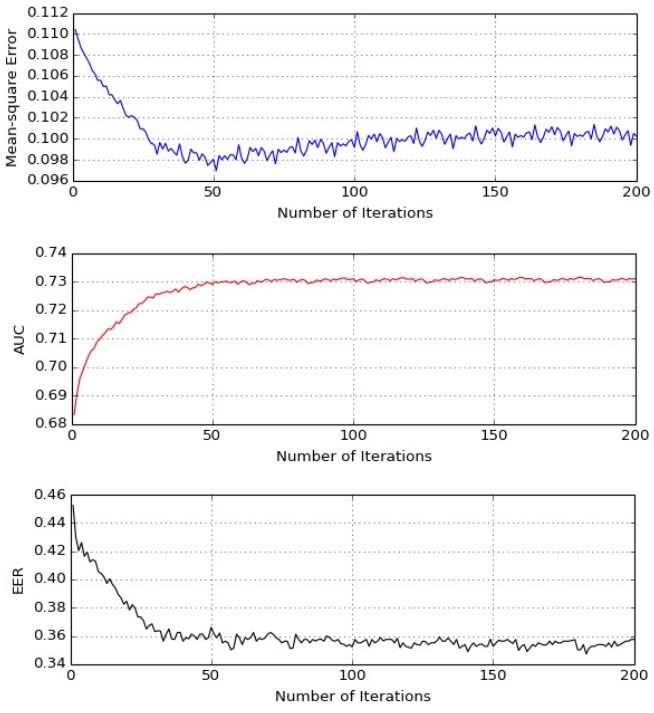

Fig. 5. Meansquare Error, AUC and EER of ANN Iterations

For a comparison with the linear method, we take two methods: the baseline(average) and the MCE objective optimization. As discussed in Section 3, the AUC objective optimization can't handle a big set of samples, thus it's eliminated in this kind of comparison. The second experiment is to investigate the performance of neural network and MCE with respect to the number of samples. We suppose that the neural network will achieve better performance with more samples. Fig. 6 shows the performance of neural network and MCE on the test set, while using a subset of all samples with different size for training. The baseline is also plotted as a horizonal line in the graph. Whatever in AUC or EER, the performance of both methods is enhanced with the accumulation of samples. Since the MCE method starts iterations from some equal weights, so that its capability is all the same as the baseline. Unlike this, the neural network method starts iterations from some random weights. Therefore when the number of samples is small such as $200,000$, the MCE method is better, but after exceeding $400,000$ iterations, the performance of the NN based method will surpass the MCE based lienar method.

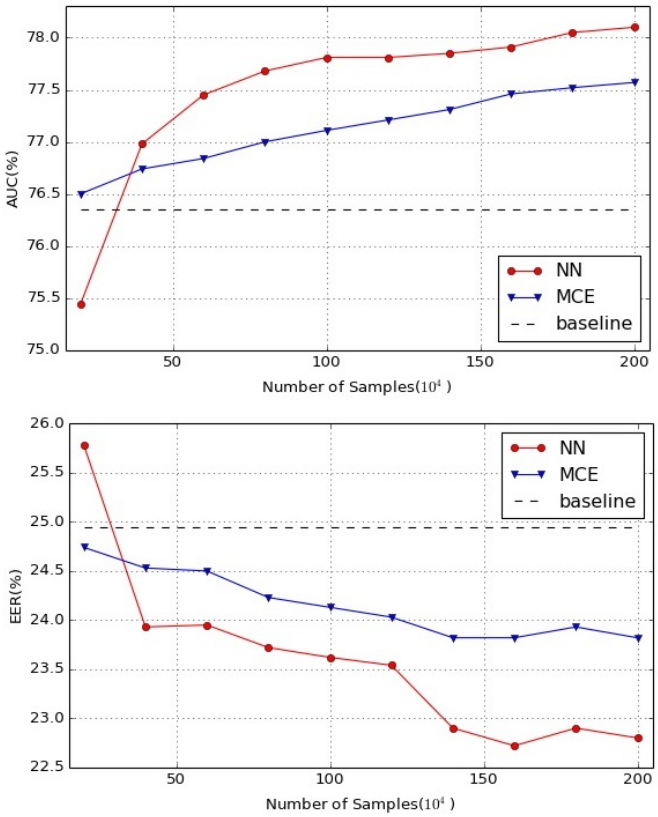

Fig. 6. AUC and EER of sample numbers

The third experiment investigates the performance of neural networks with different size on the evaluation set. The results are listed in Table I. The results show that without pre-training or different activated function, the deep layers architecture of the neural network can not be more effective than the single layer and even worse performance will be achieved. The single neuron network has no difference from a linear mapping, but its ability is weaker than the MCE method because of different objective functions. About 10

hidden neurons are enough for a tolerable performance. Based on this experiment on the evaluation set, we choose this 10 hidden neuron with single-layer as our final architecture of the neural network.

TABLE I

PERFORMANCE OF NN WITH DIFFERENT ARCHITECTURE ON EVAL SET

Layer Num	Neuron Num	AUC(%)	EER(%)
1	1	70.96	40.54
	5	73.07	35.38
	10	73.10	35.38
	20	73.05	35.87
	50	73.02	35.44
2	5	72.88	36.49
	10	72.91	36.24
	20	72.96	35.63
3	5	72.66	36.98
	10	72.90	35.52
	20	72.71	36.12

Finally, we compare the final performance of the neural network based method with two other linear methods: the baseline and the MCE objective weighted average based optimization. The ROC of three methods is as shown in Fig. 7. It can be seen that both the neural network and MCE optimization outperform the baseline significantly and the proposed method achieved a global enhancement over other two methods. The AUC, EER, and the relative improvement on EER of three methods are listed in Table II precisely. The proposed method obtain 9.4% relative improvement in EER over the baseline and achieve the best performance among three methods.

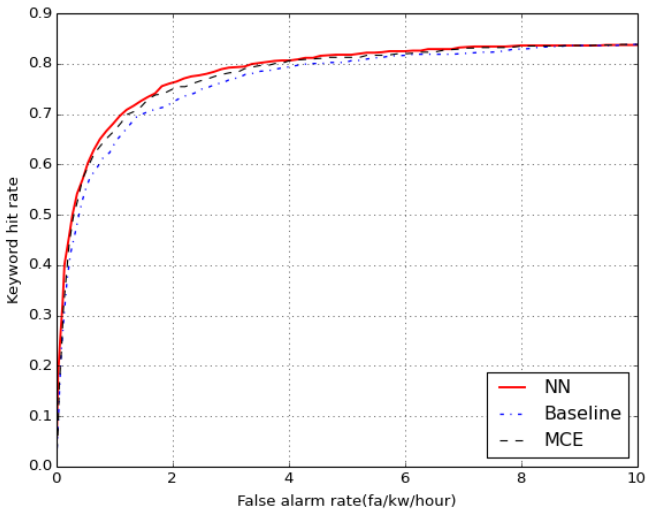

Fig. 7. Comparison of three methods on ROC

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a novel method based on the feed-forward neural network is proposed to optimize the posterior confidence measure for keyword spotting. Unlike the previous linear methods, it has an advantage that make nonlinear mapping from the phone-level confidence into the word-level.

TABLE II

PERFORMANCE OF THREE METHODS ON TEST SET

Method	AUC	EER	Relative Improvement
Baseline	76.35%	24.95%	-
MCE	77.51%	23.72%	4.9%
NN	78.17%	22.60%	9.4%

Because a phone-level confidence vector is long and sparse, activations and back-propagations are expected to be running slowly. In order to overcome this disadvantage, we propose a strategy to exploit the sparsity of confidence vectors for speeding up the method to be feasible.

In the experiments, we verify the method to be effective. Two measures are employed to evaluate the performance: AUC and EER. The performance is enhanced with the proceeding of iterations and the accumulation of samples. To be compared, two other linear methods are used: the baseline and MCE optimization. Both the proposed method and MCE method beat the baseline significantly and the proposed method based on neural network achieve the best performance among three methods.

This method is promising for enormous evolutions which have been progressed in deep learning. In the future, a more effective pre-training and fine-tuning scheme will be employed for training a feed-forward neural network with deep layers, or other activated function will be considered. It's hopeful to improve the performance of this method further more.

REFERENCES

[1] I. Szoke, P. Schwarz and P. Matejka, L. Burget, M. Karafiat, M. Fapso, J. Cernocky, "Comparison of Keyword Spotting Approaches for Informal Continuous Speech," *Proc of Interspeech*, 2005.

[2] Shilei Zhang, Zhiwei Shuang, Qin Shi and Yong Qin, "Improved Mandarin Keyword Spotting using Confusion Garbage Model," *Iternational Conference on Pattern Recognition*, 2010, pp. 3700-3703.

[3] Petr Motlicek, Fabio Valente and Igor Szoke,"Improving Acoustic Based Keyword Spotting Using LVCSR Lattices," *Proc of ICASSP* , 2012, pp. 4413-4416.

[4] H. Bourlard, B. Dhoore, and J. M. Boite, "Optimizing recognition and rejection performance in word spotting systems," *Proc of ICASSP*, 1994, Vol. 1, pp. 373-376.

[5] R. A. Sukkar and C. H. Lee, "Vocabulary Independent Discriminative Utterance Verification for Non-Keyword Rejection in Subword Based Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, Vol. 4, pp. 420-429, 1996.

[6] S. Abdou and M. S. Scordilis, "Beam search pruning in speech recognition using a posterior-based confidence measure," *Speech Communication*, vol. 42, pp. 409-428, 2004.

[7] Y. Benayed, D. Fohr, and J.P. Haton, "Confidence measures for keyword spotting using support vector machines," *Proc of ICASSP*, 2003.

[8] J.E. Liang, M. Meng, X.R. Wang, P. Ding, and B. Xu, "An improved mandarin keyword spotting system using MCE and context-enhanced verification," *Proc of ICASSP*, 2006, pp. 1145C1148.

[9] Haiyang Li, Jiqing Han, Tieran Zheng, "AUC Optimization Based Confidence Measure for Keyword Spotting," *Proc of Interspeech*, 2011.

[10] Timothy J. Hazen, Stephanie Seneff and Joseph Polifroni, "Recognition confidence scoring and its use in speech understanding systems," *Computer Speech and Language*, vol. 16, pp.49-67, Jan. 2002.

[11] Simo O. Kamppari and Timothy J. Hazen, "Word and Phone Level Acoustic Confidence Scoring," *Proc of ICASSP*, 2000, vol.3, pp. 1799-1802.