ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

Word recognition and incremental learning based on neural associative memories and hidden Markov models

Zöhre Kara Kayikci and Günther Palm

Ulm University - Institute of Neural Information Processing
89069 Ulm - Germany

Abstract. An architecture for achieving word recognition and incremental learning of new words in a language processing system is presented. The architecture is based on neural associative memories and hidden Markov models. The hidden Markov models generate subword-unit transcriptions of the spoken words and provide them as input to the associative memory module. The associative memory module is a network of binary auto- and heteroassociative memories and responsible for combining words from subword-units. The basic version of the system is implemented for simple command sentences. Its performance is compared with the performance of the hidden Markov models.

1 Introduction

We developed a hybrid system of hidden Markov models (HMMs) [1, 2] and neural associative memories (NAMs) [3] to perform the word recognition task in a language processing system for understanding simple command sentences such as "bot show apple". In the system's first step, the HMMs generate a subword-unit transcription to the auditory input. This transcription could be phonemes, demi-syllables or syllables depending on the task and the size of the available vocabulary. Then the associative memory task is to recognize words from the input stream of subword-units generated by the HMMs. This stream of words is then forwarded to a language understanding module responsible for extracting the semantics from the stream of words with respect to a set of grammatical rules [4]. In this paper, we will focus on the word recognition system.

The word recognition module is a network of interconnected auto- and heteroassociative memories using binary sparsely distributed representations on subword-unit level. The system is presently implemented to understand simple command sentences like "bot put plum (to) yellow lemon" on a small vocabulary consisting of 43 words. The learning process is initiated by a special sentence "this is X" where "X" is the novel word. While learning novel words, the HMMs generate a subword-unit transcription for the unknown word, which is used to create a new cell assembly in the corresponding heteroassociative memories. After learning, the new word can be recognized as previously stored words. The reason for using NAMs is that it is easy to add items to NAMs, whereas it is computationally expensive to generate additional HMMs for new words. The network architecture is also able to handle ambiguities that occur due to the spurious subword-units (incorrectly recognized by HMMs) in the input stream.

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

2 Neural associative memories

We have chosen the binary Willshaw model of neural associative memory [5, 6]. The patterns are stored by a "Hebbian" learning rule [7]:

$$w_{ij} = \bigvee_{k=1}^{M} x_i^k y_j^k, \tag{1}$$

where M is the number of patterns, x_k is the binary input pattern, y_k is the binary output pattern and w_{ij} corresponds to the synaptic weight of the connection from neuron i in the input population to neuron j in the address population.

Retrieving is performed by a one-step retrieval strategy with threshold:

$$y_j^k = 1 \Leftarrow \left(W x^k\right)_j = \Theta, \tag{2}$$

where the threshold Θ is set to a global value and y is the content pattern.

3 Hidden Markov models

The HMMs are used to provide the input stream of subword-units to the word recognition network, e.g. word-interval (w.i.) triphones consisting of diphones given as $p+p_R$ (first phoneme of the word with right context phoneme) or p_L-p (last phoneme of the word with left context phoneme), and triphones, defined as $p_L - p + p_R$ (central phoneme with left and right context phonemes). The topology of HMMs are three-state continuous 8-Gaussian triphone models [1, 2]. Mel Frequency Cepstral Coefficients (13 coefficients with 26 delta coefficients) are used to train HMMs [1]. The design of w.i. triphone models follows the standard Baum-Welch reestimation strategy with decision tree based triphone creation and clustering [2]. The speech corpus is composed of the training set of TIMIT speech corpus [8] and our own speech data composed of 105 different sentences, each of which spoken by 4 speakers. The models are trained with the TIMIT training set and 70 sentences for each speaker from our own speech data. For the implementation presented here, w.i. triphones are used as subword-units. Therefore, in order to get a w.i. triphone-level transcription of the auditory input, a triphone-level bigram language model is created, which is based on the TIMIT speech corpus and our own speech data.

4 Network architecture

Figure 1 shows an overview of the architecture of the word recognition network. Each box in Figure 1 corresponds to an associative memory.

The idea is that the word recognition network tries to generate a list of word hypotheses in terms of the processed w.i. triphones, each time a new w.i. triphone is read from the HMM output sequence. After processing all w.i. triphones belonging to a possible word, an output word or a superposition of matching output words is generated. All the associative memories except for HM5 consist

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

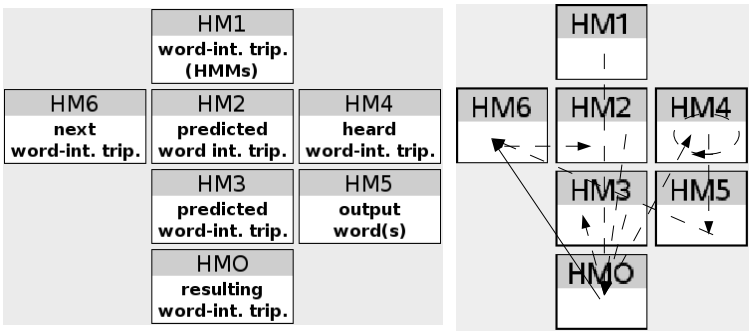

Fig. 1: Overview of the word recognition network and the connectivity within the network. The network consists of 6 interconnected associative memories and a representation area, where the memories HM1, HM2 and HM4 are autoassociative memories, while HM3, HM5 and HM6 are heteroassociative memories. The dashed arrows denote autoassociative and solid arrows denote heteroassociative connections between memories.

of n=7423 neurons, where n is the number of w.i. triphones used in HMMs. The memory HM5 contains only 200 neurons. The memories HM1, HM2 and HM4 store the w.i. triphones using 1 out of n sparse binary code vectors as input and output patterns. Although these memories have similar structures, they are used for different tasks (see below). The memory HM3, a memory matrix of $n \times n$, stores the w.i. triphone transitions within the words in the vocabulary using 1 out of n sparse code vectors. The memories HM5 and HM6 store each word in the vocabulary using two representations, i.e the phonetic transcription of the word as k out of n code vector (k is the number of w.i. triphones involved in the word) and a randomly generated sparse binary code vector. For each word, the input and output patterns in HM5 are given as phonetic transcription and as a randomly generated code vector, respectively, while the input and output patterns in HM6 are used inversely.

During retrieval, HM1 serves as an input area and presents the HMM output w.i. triphone to the network, while HM2 represents the w.i. triphone predicted by HM6 and HM3 represents the possible w.i. triphone(s) following the resulting w.i. triphone in the previous retrieval step. The outputs of the memories HM1-3 are summed up and a common threshold is then applied. In this way, the spurious w.i. triphones, which can cause ambiguities on word level, may be corrected by the network. The resulting w.i. triphone is represented in the area HMO. The memory HM4 activates the processed w.i. triphones up to the current step through the autoassociative connection with HMO and a backpropagation from itself. HM5 is responsible for generating a word hypothesis (or superposition of word hypotheses) with respect to the patterns activated in HM4. The memory HM6 predicts the w.i. triphones expected in the next step with respect to the current word hypothesis (or superposition of word hypotheses) and the resulting

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

w.i. triphone in HMO.

Now we will demonstrate how the word recognition network processes an example w.i. triphone-level trancription of the command "bot lift red ball" generated by HMMs "b+ow b-ow+t ow-t sp l+ih l-ih+f ih-f+t f-t sp r+eh r-eh+t eh-t sp ao+l ao-l sp" where "sp", which is used to determine the word boundaries, denotes "small pause" between words. Furthermore, the phonetic transcriptions for the words "red" and "ball" were incorrectly recognized by HMMs. The correct transcriptions should have been "sp r+eh r-eh+d eh-d sp b+ao b-ao+l ao-l sp".

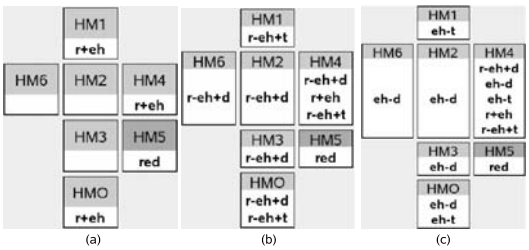

Fig. 2: The state of the word recognition network processing the w.i. triphone transcription "r+eh r-eh+t eh-t". (a) shows the processing of the first trihone "r+eh", (b) shows the processing of the second triphone "r-eh+t" and (c) shows the processing of the third triphone "eh-t".

Figure 2 shows the word recognition system which processes the w.i. triphone transcription "r+eh r-eh+t eh-t". As shown in Figure 2 (a), the assembly "r+eh" is activated in HM1, while the memories HM2 and HM3 do not receive any input at the beginning of the word. Therefore, they do not activate any output neurons. After the first w.i. triphone is activated in HM4, the word hypothesis "red" is generated in HM5 with respect to the activated w.i. triphone in HM4. In Figure 2 (b) HM6 activates the w.i. triphone expected in the next step, related to the activated word hypothesis in HM5 and the resulting w.i. triphone in HMO. Then, the second w.i. triphone has been processed and again the word hypothesis "red" is activated with respect to w.i. triphones held in HM4. In the same way the third w.i. triphone is processed in the network and the output word "red" is correctly recognized (see Figure 2 (c)).

Figure 3 shows the processing of the last transcription part "ao+l ao-l" in the HMM output. After processing the first w.i. triphone "ao+l", the network can not generate a word hypothesis due to the fact that the activated assembly in HM4 can not activate any neuron in the memory HM5, shown in Figure 3 (a). In the next step, the second w.i. triphone has been processed and a superposition of the words "ball" and "wall" is activated with respect to the w.i. triphones in HM4. Since the network can not solve the ambiguity on word level, it generates a superposition of the word patterns which will be handled on the upper level of the language processing system to resolve the ambiguity on word level using a bidirectional connection between matching verbs and objects [9].

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

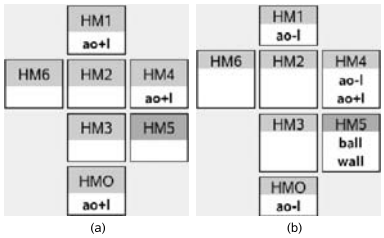

Fig. 3: The state of the word recognition network processing the w.i. triphone transcription "ao+l ao-l". (a) shows the processing of the first w.i. triphone "ao+l", (b) shows the processing of the second w.i. triphone "ao-l".

5 Incremental learning

In speech recognition applications based on HMMs, it is usually impossible to increase the vocabulary size during performance. To achieve the vocabulary enlargement, many parameter files required for the application have to be changed before the performance. However, the network architecture presented here enables us to enlarge the vocabulary by learning of new words during performance.

In our implementation, learning is triggered by a special command "this is X" where "X" is the novel word. First, HMMs preprocess the auditory input "this is X" to generate a plausible w.i. triphone sequence for the novel word. Therefore, a large bigram "language" model on subword unit level based on TIMIT sentences [8] and 105 additional simple command sentences and a large set of trained w.i. triphones are used. The basic information about the training procedure can be found in Section 3 and for more details [2]. The word recognition network uses the command "this is" to start the learning process. During this process, the learning takes place in the heteroassociative memories HM3, HM5 and HM6. In the memory HM3, the w.i. triphone transitions within the HMM output transcription for the novel word are stored using 1 out of n sparse code vectors. In the memory HM5, a new assembly is generated from randomly activated neurons and the memory HM6 generates a new assembly with respect to the w.i. triphone transcription of the novel word. Finally, the corresponding auto- and heteroassociative connections in the network are also updated. During learning, the new word is stored without updating or changing the previously stored patterns and after learning it can be used and processed as well the previously stored words. The system can correctly recognize the word "pear" in sentences like "bot show pear" after "pear" has been learned.

6 Discussion

We have presented a word recognition architecture based on neural associative memories and HMMs for a language processing system [9]. The model deals with finding out the words from an input stream of subword-units (e.g. w.i. triphones)

generated by HMMs. It is also able to represent and solve the ambiguties that occur due to the fact that the HMMs can not always generate a correct subword-unit transcription of the spoken words [9].

Compared to HMMs, the architecture has a more flexible functionality in terms of the lexicon generation. In order to enlarge the vocabulary, the modifications to the lexicon, the language model and training of new subword-unit models are necessary for HMMs, while the presented architecture needs only a subword-unit level representation from HMMs for the novel word without further training of HMMs (i.e., the lexicon is automatically updated by the associative memories in the system with respect to the transcription of the new word). In addition, the associative memories need only one training step for the new pattern without affecting previously stored patterns.

Due to the high storage capacities of the sparse binary associative memories [3], the presented architecture scales well with vocabulary size.

The recognition performance is evaluated on a small vocabulary of 43 words. The test set is composed of 35 simple command sentences from 4 speakers and there are totally 504 words in the test set. The presented system recognized 98% of the words in the test set, whereas HMMs yielded 96% [9].

The speed of the system was analyzed on a standard computer (Pentium 4, 2.66 MHz) with comparison to a word recognition application based on word-level HMMs. It takes 7 seconds for the standard HMM based application to recognize a sentence like "bot show pepper", whereas it is measured as 6 seconds for the presented system. It also takes 6 seconds for the system to learn a new word in a sentence like "this is pear" where "pear" is the novel word, which comprises the generation of the new transcription by HMMs, recognition of the words in the spoken sentence and learning of the new word "pear" by the associative memories.

References

[1] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*, Prentice-Hall, Inc., Upper Saddle River, 1993.

[2] S. Young, et al., *The HTK book for HTK version 3.2.1*, Cambrige University, Engineering Department, 2002.

[3] G. Palm, On associative memory, *Biological Cybernetics*, 36:19-31, 1980.

[4] H. Markert, A. Knoblauch, and G. Palm, Modelling of syntactical processing in the cortex,*BioSystems*, 89:300-315, 2007.

[5] D. Willshaw, O. Buneman and H. Longuet-Higgins, Non-holographic associative memory, *Nature*, 222:960-962, 1969.

[6] J. Buckingham and D. Willshaw, Performance characteristics of the associative net, *Network*, 3:407-414, 1992.

[7] D.-O. Hebb, The organization of behaviour, John Wiley, Newyork, 1949.

[8] TIMIT Acoustic-Phonetic Continuous Speech Corpus, National Institute of Standartsand Technology Speech Dics 1-1.1, NTIS Order No. PB91-505065, 1990.

[9] Z. Kara Kayikci, H. Markert and G. Palm, Neural Associative Memories and Hidden Markov Models for Speech Recognition, proceedings of the 20^{th} *international joint conference on neural networks* (IJCNN 2007), August 12-17, Orlando, Florida (USA), 2007.