Applying Convolutional Gated Recurrent Deep Neural Network for keywords spotting in continuous speech

Hinda DRIDI, Kais OUNI

Abstract— Recently, the task of Keywords Spotting (KWS) in continuous speech has known an increased interest. It has been considered as a very challenging and forward-looking field of speech processing technologies. The KWS systems have been widely used in many applications, like, spoken data retrieval, speech data mining, spoken term detection, telephone routing, etc. In this paper, we propose a two-stage approach for keywords spotting. In first stage, the inputted utterances will be decoded into a phonetic flow. And in second stage the keywords will be detected from this phonetic flow using the Classification and Regression Trees (CARTs).

The phonetic decoding of continuous speech is largely taking benefits of deep learning. Very promising performances have been achieved using Deep Neural Network (DNN), Convolutional Neural Network (CNN) and other advanced Recurrent Neural Networks (RNNs) like Long Short Term Memory (LSTM) and Gated Recurrent units (GRU). This paper builds on these efforts and proposes potentially more pertinent architecture by combining CNN, GRU and DNN in a single framework that will be called “Convolutional Recurrent Gated Deep Neural Network” or simply “CNN-GRU-DNN”. The work will be conducted on TIMIT data set.

Index Terms— Deep Neural Network, Convolutional Neural Network, Recurrent Neural Network, Long Short Term Memory, Gated Recurrent units, Keywords Spotting, two-stage approach, Classification and Regression Tree, TIMIT.

1 INTRODUCTION

In last few years, due to the significant technological advances a large amount of spoken data has been easily stored, shared and accessible in Internet and in different datasets every day. However, supervising these large quantities is not an easy task for human being. Therefore, it’s necessary to develop new technologies and tools for an effective access to these amounts of data to extract all useful and pertinent information contained therein. In this context, the keywords spotting (KWS) task has attracted the attention of research and industry communities.

The KWS task is a very forward-looking field of speech processing technology, which aims to detect and identify some pre-defined words (keywords) in utterances of continuous speech. It has been widely used in many applications like, spoken document retrieval, spoken term detection, spoken data mining, telephone routing, etc. Despite the challenging nature of keywords spotting, several systems have been developed over the years and have achieved interesting results. Recently, thanks to the advanced algorithms available for training neural networks and especially deep networks more promising performances have been obtained.

The aim of this work is to develop an efficient system for keywords spotting. This system is based on a two-stage approach. In first stage, we propose a very deep learning architecture for decoding the inputted utterances into phonetic flow. Then, the predefined keywords will be detected from this phonetic flow using the Classification and Regression Trees (CARTs).

The deep learning architecture proposed for phonetic decoding, is inspired by the model presented in [11]. This proposed architecture will be composed by combing three deep neural networks: Convolutional Neural Network (CNN), Gated Recurrent Units (GRU) and Deep Neural Network (DNN) in a single framework architecture that will be called in this paper “Convolutional Recurrent Gated Deep Neural Network” or simply “CNN-GRU-DNN”. These three subnetworks will interact with each other in a unified architecture to generate a phonetic transcription of continuous speech.

The remainder of this paper will be organized as follows. Influential works related to the keywords spotting task are presented in section 2. A main description of our keywords spotting system, the most performing deep architectures used for phonetic decoding and the process of building our proposed deep architecture are presented in Section 3. The metrics used for evaluating the keywords spotting system are presented in section 4. The experiment setup and results are described in Section 5. Finally, the conclusion with some directions for future work is outlined in Section 6.

2 RELATED WORKS TO KEYWORDS SPOTTING

Recently, researches related to keywords spotting in continuous speech have known an increased interest. Several approaches have been investigated in literature over the years. Some of the earliest ones were based on Dynamic Time Warping (DTW), in this approach the keywords are searched by computing an alignment distance between a template representing the target keyword and all segments of the test speech signal to efficiently find a match. This approach has been used by many researchers but along with the evolution of speech processing technologies it started showing its shortcomings.

[1], [2], [5]

Consequently, other techniques were investigated later, where the most popular ones were based on Hidden Markov Models (HMMs). In this approach, the keywords were represented by their phonetic transcriptions models and the non-keywords were represented by filler or garbage models. For a given utterance, this KWS technique outputs a sequence of keywords and non-keywords. Using filler models was efficient to model the extraneous speech and to bring more promising results compared with the template-based approach. Further ameliorations to HMM-based techniques have been achieved using neural networks like; Reccurent Neural Networks (RNNs), Time-Delay Neural Networks (TDNNs), and more recently Deep Neural Networks (DNNs). However, the major inconvenient of HMM-based KWS technique that a simple change in the application vocabulary imposes to restart the recognition stage again, which is a very time-consuming. [3], [4], [6]-[8]

To overcome this problematic, a two-stage approach has been recently proposed. This KWS approach performs on two main steps that are, indexing and search. In first step, the inputted utterance will be processed with automatic speech recognition (ASR) system to get a phonetic transcription. This phonetic transcription serves to obtain an intermediate representation, known as the index. In second step, namely keywords search, the keywords occurences will be simply and quickly searched using this index. The two-stage approach allows much faster detection because the main and hard stage of KWS task has been done beforehand, without needing any prior knowledge of the keywords to be searched. And if we want to search new keywords, only the second stage will be done again. [7]- [8]

3 The proposed Keywords Spotting system

Motivated by interesting performances of the two-stage approach we will adopt it to propose our keywords spotting system. This proposed KWS system will perform as following: in first stage, a phonetic transcription of the inputted utterance will be done using our proposed deep learning architecture in combination with the Hidden Markov Models, and in second stage, the predefined keywords will be detected from this phonetic flow using the Classification and Regression Trees (CARTs).

Our contributions in this paper will be proposing an ameliorated deep architecture to get better phonetic decoding in the first stage of our KWS system and proposing an efficient use of the Classification and Regression Trees for searching keywords in the second stage.

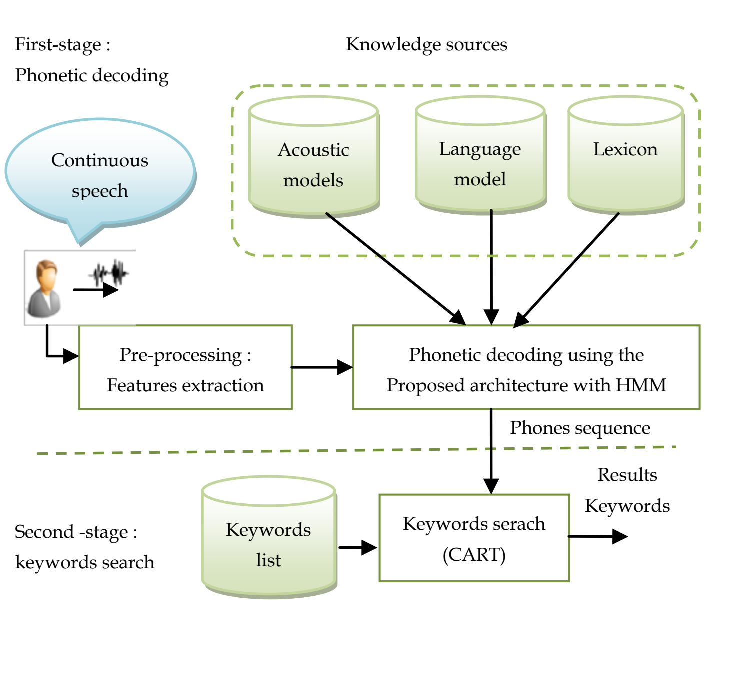

Fig. 1. The two-stage proposed Keywords Spotting system.

3.1 First stage: Phonetic decoding

Decoding continuous speech into phones sequence is a fundamental task used in many automatic speech recognition tasks. It has been treated by several ways, where the earliest one was the classic GMM-HMM model obtained by combining Gaussian Mixture Model (GMM) and Hidden Markov Model (HMM). The phone recognition rates obtained with such architecture were acceptable but not very efficient. In last few years, with the tremendous success of artificial intelligence techniques much deep architectures have been introduced, in the following sub-section we will present some of the most promising ones.

3.1.1 An overview of the most promising deep architectures

Some of the most popular deep architectures investigated for speech and phone recognition is Deep Neural Network (DNN). It's a conventional Multi-Layer Perceptron (MLP), with multiples layers of hidden units. All hidden units of one layer are connected to those in next layer using unidirectional connections. The first layer in DNN will extract new representation from the speech input, and then its output will be passed to a next layer in order to generate a new higher representation, and so on; for all next layers until reaching the top layer. A DNN may be used in HMM-based speech recognition system and be trained either in a supervised way or unsupervised one using the pre-training approach introduced by Hinton et al. in [14].

Mohamed et al [26] proposed the first application of pre-trained DNN for phone recognition. Their model has achieved a phone error rate of 22.4% on TIMIT benchmark (test set), which has been confirmed notably outperforming traditional "GMM-HMM" models and giving promising results for practical use.

- *Dridi Hinda*
  Research Unit Signals and Mechatronic Systems, UR13ES49
  National Engineering School of Carthage, ENICarthage
  University of Carthage, Tunis, Tunisia
  hinda1.dridi@gmail.com
- *Kais OUNI*
  Research Unit Signals and Mechatronic Systems, UR13ES49
  National Engineering School of Carthage, ENICarthage
  University of Carthage, Tunis, Tunisia
  kais.ouni@enicarthage.rnu.tn

Recently, more advanced alternatives of neural networks have been investigated like the Convolutional Neural Network (CNN). It contains a pair of convolution and pooling layers followed by several fully connected layers. The convolution layer performs convolution operations to get outputs from small local regions called receptive fields. In this layer, the neurons are organized into feature maps, where those belonging to the same feature map share same weights or filters. Each convolution layer will be generally followed by a pooling layer, which is also organized into a same number of feature maps as the convolution layer but with smaller maps.

Convolutional Neural Networks have been widely used for speech recognition in many works. Abdel-Hamid et al [15] proposed a CNN with new limited-weight-sharing scheme and with frequency convolution. The experimental results of their CNN-HMM model have achieved a phone error rate of 20.36% on TIMIT dataset, which has improved more the rates of phone recognition compared with DNN. In other work proposed by Loth [17] a CNN-HMM architecture with convolution over both time and frequency has been introduced. The phone error rate reported in this work on TIMIT dataset was 16.7%.

A main weakness of convolutional neural network is that it can just model a limited temporal dependency. To overcome this problem, some researchers have investigated the use of recurrent neural network (RNN) for speech recognition. Nevertheless, training RNN is a difficult task, which may lead to many problems of gradient vanishing and exploding. Consequently, others alternative architectures of RNN have been introduced where the most promising ones are Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU).

In a standard LSTM network the reccurent hidden layer has a number of recurrently connected units called “memory blocks”. Each memory block contains one or more self-connected memory cells to store the contextual information and three multiplicative gates called input, output and forget gate to control the flow of information.

For further improvements the Bidirectional LSTM (BLSTM) may be used. It takes benefit of both past and future contexts in speech signals. A BLSTM is composed by a forward layer to process the input sequence in the forward direction and a backward layer to process the input sequence in the backward direction. The resulted output is obtained by concatenating the outputs of the two layers.

Some of the earliest works introducing Long Short Term Memory (LSTM) for speech recognition was presented by Graves et al [23]. They have shown that bidirectional LSTM (BLSTM) are more promising than unidirectional LSTM. They have achieved a phone error rate of 17.7% on TIMIT dataset by using a deep BLSTM.

To ameliorate the computational efficiency of LSTM an alternative architecture called Gated Recurrent Units (GRU) has been introduced. This architecture is an advanced variant of RNN, which allows also solving the gradient vanishing problem but with using a less number of weights. Compared with LSTM, Gated Recurrent Units is simpler. It contains only two multiplicative gates; update and reset, where the “update gate” is obtained by combining the “forget” and the “input” gates. GRU manages the flow of information inside the units without needing a separate memory cell. Similarly to LSTM, the Gated Recurrent Units has been also used in a bidirectional alternative. [29]

More recently, a very improved deep architecture called “Convolutional Long short-Term Memory Deep Neural Network” (CLDNN) [11] has been introduced to achieve higher recognition accuracy. This architecture combines in a single framework CNN, LSTM and DNN.

The CLDNN model has been used in HMM-based speech recognition architecture in the work proposed by Sainath et al. [11] and has achieved a WER of 17.3% on a large vocabulary task, which has brought a 4% relative improvement over the LSTM for the same task.

3.1.2 The proposed architecture

Inspired by the work presented in [11] we propose an ameliorated deep architecture, which will be called “Convolutional Gated Recurrent Deep Neural Network”. This proposed model is built by combining CNN, GRU and DNN in a single framework that will be defined like “CNN-GRU-DNN”.

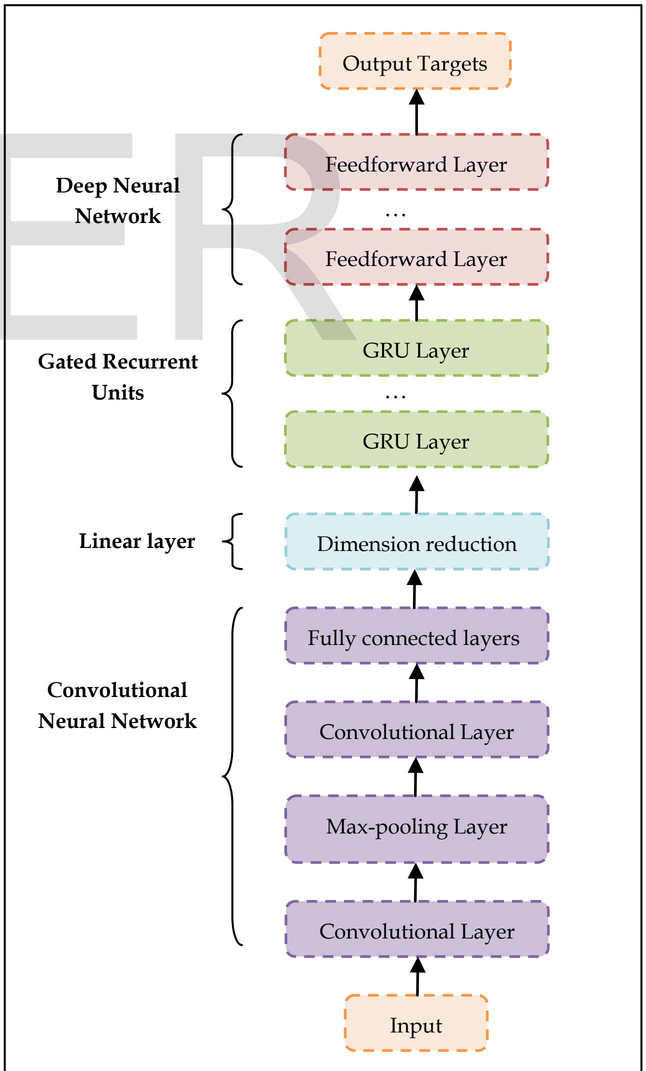

Fig. 2. An illustration of the proposed deep architecture.

The proposed model takes benefit of the distinct strength of

CNN, GRU and LSTM and reduces the effects of their individual shortcomings.

The baseline "CNN-GRU-DNN" is shown in Fig.2, for this proposed architecture we feed the input features within their temporal context into a Convolutional Neural Network to reduce the spectral variations existing in speech signals. In this work, the CNN used is only with convolution along frequency. The dimension of the last layer in this CNN architecture is very large; for that a linear layer is added after these CNN layers. This linear layer reduces the number of parameters without deteriorating the recognition performances. Next, the output of this linear layer is passed into a stack of GRU layers to model the long-term temporal dynamics. Finally, after performing frequency and temporal modeling, we pass the output of the top GRU layer into a DNN composed by few number of fully-connected feed-forward layers to provide a better discrimination of the output targets.

The "CNN-GRU-DNN" outputs a probability distribution over the possible labels of the central frame. To obtain the phones sequence, these probabilities must be divided by the HMM states produced by the higher DNN layer, and then will be passed to a Viterbi decoder.

3.2 Second stage: keywords search

As we detailed previously, the first stage of our proposed KWS system will serve to generate a phonetic transcription of the inputted utterances. And in second stage, the predefined keywords will be detected from this phonetic flow using the Classification and Regression Tree (CART).

The CART is a machine learning algorithm used to predict how a given input may lead to a specific output according to some contextual factors. The CART algorithm is defined by a number of yes or no questions associated to each non terminal node. A new branch conducting to the next question will be presented for each possible answer. All these answers will build a tree like structure, where the terminal nodes called also leaves maintain the specific output.

The CART building requires a training set composed by inputs (features) and their corresponding labels (outputs). For our keywords spotting task the CART algorithm is used to retrieve the grapheme corresponding to a given phoneme. First, for training the spelling and pronunciations of words existing in training set must be aligned. These alignments will produce different grapheme-phoneme correspondences; in such a way each phoneme in a word will get its corresponding grapheme (we may use the null graphemic or phonemic symbol "_" when the spelling and pronunciation of a word are with different lengths). All these correspondences will be used to train a number of classification and regression trees, one tree for each phoneme.

According to the value of the contextual factor, assigned to each non terminal node, the tree will be so crossed starting from a parent node and then passing to a child one, until reaching the terminal node (leaf). The grapheme corresponding to a given phoneme is maintained by this leaf.

The node splitting and tree design are based on the minimum entropy criterion that can be defined as: [34]

$$H(L\,|\,node) = -\sum_{l\in L} P(l\,|\,node)\ \log_2 P(l\,|\,node) \qquad (1)$$

where L are the possible graphemes, and $P(l\,|\,node)$ is the grapheme's occurrences for a given node.

According to the phoneme and grapheme contexts the parent node will be splitted into two child nodes. We can define the average entropy as: [34]

$$H(L\,|\,child_1, child_2) = H(L\,|\,child_1)P(child_1) + H(L\,|\,child_2)P(child_2) \qquad (2)$$

where $P(child_1)$ and $P(child_2)$ refer to the probabilities of reaching the two child nodes i.e, the probabilities for which the contextual factor C_i falls respectively, into C_1^j and C_2^j, which are the two subsets obtained by the partition C_i^j of the values of C_i, where i and j denote the indexes presenting the phonemes and the partition of theirs values.

The best splitting $C_{i,best}^{j,best}$ presents the maximum difference between the entropies values before and after splitting. This difference is defined as $I(L,\mathrm{C}_i^j)$ and it corresponds to the average of mutual information between the graphemes to predict and the splitting: [34]

$$I(L,C_i^j) = H(L\,|\,parent) - H(L\,|\,child_1, child_2) \qquad (3)$$

To obtain the best splitting $C_{i,best}^{j,best}$, we must get first the partition $C_i^{j,best}$ that maximizes $I(L,C_i^j)$ and then we maximize $I(L,\mathrm{C}_i^{j,best})$.

$$C_i^{j,best} = \arg\max_j I(L,\mathrm{C}_i^j) \qquad (4)$$

$$C_{i,best}^{j,best} = \arg\max_i I(L,\mathrm{C}_i^{j,best}) \qquad (5)$$

These steps must be repeated for all the child nodes until obtaining a maximum of $I(L,\mathrm{C}_i^j)$ inferior to a predefined threshold, for which the entropy reduction is considered insignificant.

During test stage, the different CARTs trained for each phoneme will be used to retrieve the predefined keywords. Each keyword is detected using the phones sequence, outputted by the proposed deep model described previously, and by selecting at a time the phoneme on left and on right of the current phoneme. The CART of this phoneme will be crossed from the root node until reaching the leaf, the corresponding grapheme is maintained so by this leaf. These steps will be repeated with the following phoneme and its corresponding grapheme will be retrieved similarly. Finally, the detected keyword is obtained by concatenating all the predicted graphemes that have been obtained.

4 Evaluation metrics for the KWS system

Once we have tested the keywords spotting system and obtained the putative keywords occurrences, these results will be evaluated. Generally, the errors generated by a KWS system may be explicated by two different scenarios. The first one is when the KWS system doesn't detect a keyword, which is already pronounced in the inputted utterance. And the second one is when the KWS system detects a keyword, which is actually not pronounced in the inputted utterance. The first error is called "missing" and the second error is called "false alarm". [7], [8]

- Missed Detection Rate (missing): For a given keyword, q, it can be defined as:

$$P_{miss}(q) = \frac{N_{miss}(q)}{N_{True}(q)} \tag{6}$$

where $N_{miss}(q)$ corresponds to the number of missed detections and $N_{True}(q)$ corresponds to the number of reference occurrences.

The rate of total missed detections produced by the KWS system may be computed as:

$$P_{miss} = \frac{1}{K}\sum_{q=1}^{K}\frac{N_{miss}(q)}{N_{True}(q)} \tag{7}$$

where K corresponds to the number of keywords

- Detection Rate or accuracy: corresponds to the number of references keywords occurrences, which are correctly detected by the KWS system. For a given keyword, q, the detection rate can be defined as:

$$P_{correct}(q) = 1 - P_{miss}(q) \tag{8}$$

- False Alarm Rate: For a given keyword, q, the false alarm rate can be defined as:

$$P_{FA}(q) = \frac{N_{FA}(q)}{N_{NT}(q)} \tag{9}$$

where $N_{FA}(q)$ corresponds to the number of false alarms and $N_{NT}(q)$ corresponds to the non-target trials.

The overall rate corresponding to the total false alarms produced by the KWS system can be calculated as:

$$P_{FA} = \frac{1}{K}\sum_{q=1}^{K} P_{FA}(q) \tag{10}$$

- There are two others measures that can be taken as evaluation metrics for the KWS system, which are recall and precision rates.

The recall rate may be defined in terms of the number of total detections to make $N_{True}(q)$ and the keywords correctly detected $N_{correct}(q)$.

$$\text{Recall} = \frac{100{*}N_{correct}(q)}{N_{True}(q)} \tag{11}$$

The precision rate may be defined in terms of the number of keywords correctly detected $N_{correct}(q)$, and the number of false alarms $N_{FA}(q)$.

$$\text{Precision} = \frac{100{*}N_{correct}(q)}{N_{\text{correct}(q)}+N_{FA}(q)} \tag{12}$$

Once we have presented our basic approaches for phonetic transcription and for keywords search, as well as presenting the mains evaluation metric of the KWS system we present in next section the experiments setup and the obtained results.

5 Experimental Setup and Results

5.1 Experimental results of phone recognition on TIMIT

The TIMIT dataset consists of 6,300 sentences recorded by 630 speakers of 8 major dialects of American English. By removing the SA sentences (two sentences recorded by all the speakers), we get a training set containing 3,696 sentences from 462 speakers. A test set containing 192 sentences from 24 speakers. In order to validate our results and to adjust the network parameters, a random 10% of the training set, which contains 400 sentences from 50 speakers, was held out and taken as development (dev) set.

5.1.1 Baselines

In all the experiments, a bigram language estimated from the training set was used. All training labels are obtained through forced alignment using a well trained "GMM-HMM" model with 1946 tied context dependent HMM states. During decoding step the phoneme label outputs were mapped to the usual set of 39 labels. We used Kaldi [32] for feature extraction, decoding, and training of the initial "GMM-HMM" model and all the baselines neural networks.

First, we investigate the CNN, GRU and DNN models used in this paper. The CNN is trained with two convolution layers of 128 and 256 filters, respectively and using limited weight sharing scheme (LWS). The first convolution layer is followed by a max-pooling layer with a pooling size of 6 and a sub-sampling factor of 2, while no pooling was used for the second layer. After these convolution-pooling layers four fully connected layers are added, each of them with 1024 hidden units.

In first step, we will use two GRU layers each of them with 1024 units. They are trained using the truncated back-propagation though time (BPTT) learning algorithm.

The DNN is composed by a few number of fully-connected feed-forward layers, trained in a supervised way. These layers are all with 1024 hidden units and sigmoid activation function.

For training CNN and DNN the stochastic gradient decent (SGD) is used. We tried to choose specific values for the initial and final learning rates to get a stable convergence. For fine-tuning, the initial learning rate is set to 0.0004 for the CNN and to 0.008 for the DNN. This learning rate will be divided by two for each increasing in cross-validation frame accuracy for a single epoch less than 0.5%. For these experiments the SGD uses mini-batches of 256 frames.

The input to all these networks are 25 ms frames of 40-

dimensional filterbank features (FBANK features), along with their first and second temporal derivatives, computed every 10 ms.

Table 1 illustrates the phone error recognition (PER) rates obtained for the different neural networks (CNN, DNN and GRU) described previously.

TABLE 1
PHONE ERROR RATES OBTAINED FOR THE TIMIT DATASET WITH VARIOUS ARCHITECTURES

Method	PER % (dev core)	PER % (test core)
DNN (6 layers)	20.45	21.18
CNN	17.43	18.83
GRU (2 layers)	17.52	18.85

We observe that more improved phone recognition performance has been obtained using CNN over DNN. This significance can be explicated by the invariance of CNNs to small frequency shifts, which normally occur in speech signals. That makes the CNNs more robust to speaker variations than the DNNs. From these results we can also observe that the two GRU layers are outperforming the traditional DNNs and giving comparable performances to CNNs. More interesting improvements in performances of the GRU based architecture over the CNNs may be obtained by increasing the number of GRU layers in the stack.

5.1.2 The proposed architecture

In this section, we evaluate different combinations of CNN, DNN and GRU models, to justify our choice of the proposed architecture. First, we add 2 GRU layers after the CNN; this combination is defined like "CNN-GRU". Then, we add a DNN composed of 3 fully-connected feed-forward layers after the 2 layers of GRU; this combination is defined like "GRU-DNN".

Finally, the proposed architecture will be defined like "CNN-GRU-DNN". This architecture performs in three steps: first, the input features are passed into a CNN, and then the output of the CNN layers is passed into a linear layer to reduce the number of parameters. After this linear layer two GRU layers are added. Finally, after performing frequency and temporal modeling, the top GRU layer output is passed into 3 DNN layers. All results are reported in table 2.

The performances obtained by the proposed model "CNN-GRU-DNN" can bring up to 0.97% improvement in the recognition rates over a GRU model used alone for the dev set, and up to 0.89% for the test set. This improvement is not surprising due that this proposed model take benefits from the complementary individual modeling capabilities of the three neural networks (CNN, GRU and DNN), and is demonstrated to be more effective than each of all these subnetworks used alone.

TABLE 2
PHONE ERROR RATES OBTAINED FOR THE TIMIT DATASET USING THE PROPOSED ARCHITECTURE

Method	PER % (dev core)	PER % (test core)
GRU	17.52	18.85
CNN-GRU	16.77	18.22
GRU-DNN	17.21	18.60
CNN-GRU-DNN	16.55	17.96

To provide a proper performance evaluation, we make a set of experiments to compare the results of our proposed model "CNN-GRU-DNN" with the CLDNN model introduced in [11]. This later model is composed by combining CNN, LSTM and DNN in a single framework defined like "CNN-LSTM-DNN". In our experiments for the CLDNN architecture we use the same CNN, linear layer and DNN models as our proposed architecture and we use two LSTM layers, each of them with 1024 memory cells. These LSTM layers are also trained using the truncated back-propagation though time (BPTT) learning algorithm.

TABLE 3
COMPARISON BETWEEN THE PROPOSED MODEL AND THE CLDNN MODEL

Method	PER % (dev core)	PER % (test core)
LSTM (2 layers)	17.76	18.97
CLDNN	16.77	18.10
GRU (2 layers)	17.52	18.85
CNN-GRU-DNN	16.55	17.96

We observe that the performances of LSTM and GRU layers are comparable with a bit advance for the GRU ones. Also, these reported phone recognition rates confirm that using GRU instead of LSTM in the proposed architecture has significantly improved the performances compared to the CLDNN architecture, while having less number of parameters. The performances obtained by our proposed model "CNN-GRU-DNN" can bring up to 0.22% improvement in the recognition rates over the CLDNN model for the dev set, and up to 0.14% for the test set.

5.1.3 Further experiments for the proposed model

In this section, we try to investigate additional modifications for the proposed model, experimented previously, to further improve the performances.

First, we suggest using two different pooling operations, namely max and average pooling for the CNN composing our proposed model. The two other subnetworks are kept the same as previous experiments.

TABLE 4

PHONE ERROR RATES OBTAINED FOR THE TIMIT DATASET FOR THE PROPOSED MODEL USING DIFFERENT POLLING STRATEGIES IN CNN

Method	PER % (dev core)	PER % (test core)
Max	16.55	17.96
Average	16.84	18.28

We observe that the max-pooling function performs better than the average-pooling function. The max-pooling function has the ability to emphasize the transients, in contrary of the average one which smooths them out. Consequently, in all the following experiments, the CNN model used in the proposed architecture will be with a max-pooling function.

We pass now to show how the depth (number of layers) of GRU may affect the overall performance of the proposed deep model. A set of experiments is conducted using respectively 2, 3 and 4 GRU layers, all parameters of the others architectures are kept the same as previously.

TABLE 5

PHONE ERROR RATES OBTAINED FOR THE TIMIT DATASET WITH DEEPER GRU LAYERS IN THE PROPOSED MODEL

Number of GRU layers	PER % (dev core)		PER % (test core)	
	GRU	CNN-GRU-DNN	GRU	CNN-GRU-DNN
2 layers	17.52	16.55	18.85	17.96
3 layers	17.17	16.21	18.49	17.65
4 layers	16.64	15.77	17.90	17.19

These results show that deeper "CNN-GRU-DNN" models may bring further improvements in the phone recognition performances. The lowest error rates are obtained using 4 GRU layers, however increasing the number of GRU layers beyond that make the training hard and seems to complicate the training without bringing consistent improvements.

The proposed model using unidirectional GRU layers either shallow or deep one has shown very interesting phone recognition rates addressed on TIMIT task. In next experiments we are interesting about ameliorating further the performances and using bidirectional GRU (BGRU) layer instead of unidirectional one in the proposed model. This bidirectional GRU is composed for each depth by two unidirectional GRU layers: a forward and backward layer, each of them with 512 units. For training the bidirectional GRU layer, the context sensitive-chunk BPTT (CSC-BPTT) algorithm is used.

In this case, the proposed architecture will be defined like "CNN-BGRU-DNN". This architecture performs in three steps: first, the input features are passed into a CNN, and then the output of the CNN layers is passed into a linear layer to reduce the number of parameters. After this linear layer two BGRU layers are added. Finally, after performing frequency and temporal modeling, the top BGRU layer output is passed into 3 DNN layers. All parameters of the others architectures are kept the same as previous section.

TABLE 6

PHONE ERROR RATES OBTAINED FOR THE TIMIT DATASET USING THE PROPOSED ARCHITECTURE WITH BGRU LAYERS

Method	PER % (dev core)	PER % (test core)
BGRU (2 layers)	16.94	17.87
CNN-BGRU-DNN	16.03	17.15

These results show that the proposed model using Bidirectional GRU layers may bring further improvements in phone recognition performances over the model using unidirectional GRU layers. This efficiency is not surprising due to the ability of bidirectional GRU layer in exploiting the bidirectional contextual information (previous and future context), contrariwise to the unidirectional GRU layer that can exploit only the past history.

The performances obtained by the proposed model "CNN-BGRU-DNN" can bring up to 0.91% improvement in the recognition rates over a BGRU model used alone for the dev set, and up to 0.72% for the test set.

Now, we suggest comparing the results obtained with our proposed model "CNN-BGRU-DNN" using two BGRU layers and the results of the CLDNN model using also two BLSTM layers, each of them with 512 memory cells per direction (forward LSTM and backward LSTM). These BLSTM layers are also trained using context sensitive-chunk BPTT (CSC-BPTT) algorithm. All the parameters of others architectures are kept the same as previously.

TABLE 7

COMPARISON BETWEEN THE PROPOSED MODEL "CNN-BGRU-DNN" AND THE CLDNN MODEL

Method	PER % (dev core)	PER % (test core)
CLDNN	16.43	17.61
CNN-BGRU-DNN	16.03	17.15

These results confirm that the phone recognition rates obtained with the proposed model "CNN-BGRU-DNN" are more interesting than the CLDNN model. Using BGRU layers instead of BLSTM layers is more performing. The performances obtained by the proposed model can bring up to 0.4% improvement in the recognition rates over the CLDNN model for the dev set, and up to 0.46% for the test set.

We pass now to show how the depth of BGRU layers may affect the overall performance of the proposed model. A set of experiments is conducted using respectively 2, 3 and 4 BGRU layers.

TABLE 8

PHONE ERROR RATES OBTAINED FOR THE TIMIT DATASET USING BGRU LAYERS IN THE PROPOSED MODEL

Number of	PER % (dev core)	PER % (test core)

BGRU layers	BGRU	CNN-BGRU-DNN	BGRU	CNN-BGRU-DNN
2 layers	16.94	16.03	17.87	17.15
3 layers	16.48	15.67	17.42	16.56
4 layers	16.04	15.21	17.10	16.19

These results confirm that increasing the number of BGRU layers can bring more improvements in phone recognition rates. For our experiments the best number of BGRU layer to add was four. A deep BGRU provide an efficient way to model the long-range history and the non-linear relationship structures. The proposed deep architecture helps to further reduce the PER and to give promising recognition results when adding more BGRU layers. Theoretically, increasing the number of BGRU layers in the proposed model may not harm, while practically that will let the convergence more slow and the network may broke after few epochs.

In last step of our work and to bring more improvements for the proposed model performances, we introduce a set of experiments using the proposed model with four BGRU layers and different features types. The used features are 39 dimensional MFCC features, 40 dimensional filterbank features and the LDA+STC+FMLLR features.

These later features are obtained by splicing 11 frames (5 on the left and right of the current frame) of 13 dimensional MFCCs; then we apply a linear discriminant analysis LDA to reduce the dimension to 40. The MFCCs are normalized with cepstral mean-variance normalization (CMVN). After that, the semi-tied covariance (STC) transform is applied on the previous features. Finally, we apply on these features speaker adaptation using the feature-space maximum likelihood linear regression (FMLLR).

TABLE 9

PHONE ERROR RATES OBTAINED FOR THE TIMIT DATASET USING "CNN-BGRU-DNN" MODEL WITH DIFFERENT FEATURES TYPES

Features	PER % (dev core)	PER % (test core)
MFCC	15.63	16.58
FBANK	15.21	16.19
FMLLR	14.69	15.72

We find that the proposed model with four BGRU layers and using FMLLR features achieve a phone error rate of 15.72% for the TIMIT test set which is the most promising and performing result obtaining in this paper. Making a comparison with the CLDNN model confirms that our proposed model reached the highest phone recognition rates and achieved more improved performances.

5.2 Experimental results of the proposed KWS system

In this section we present the experiments and results of our keywords spotting system, using the phones sequences generated with the proposed deep architecture (using four BGRU layers and FMLLR features) combined with Hidden Markov Models (HMMs) in a merge Context dependent hybrid architecture.

In our experiments we selected forty-two words as keywords from the TIMIT dataset. These keywords can be partitioned into three groups: Short-length Keywords Group (SKG) containing the keywords with 4, 5 and 6 phones. Medium-length Keywords Group (MKG) containing the keywords with 7, 8 and 9 phones. Finally, Long-length Keywords Group (LKG) containing the keywords with more than 9 phones.

TABLE 10

RESULTS OF KEYWORDS SPOTTING FOR DIFFERENT WORDS LENGTH USING A TWO-STAGE APPROACH

Keywords group	Short-length Keywords	Medium-length Keywords	Long-length Keywords
Search terms	14	14	14
Reference occurrences	109	133	102
Missed Detection Rate	0.43	0.25	0.17
Correct Detection Rate	0.56	0.74	0.82
False alarm rate	0.40	0.11	0.01
Precision %	58.49	89.71	97.67
Recall %	56.88	72.18	82.35

From the results listed in the following table we observe that the highest detection rate occurs for detecting the long-length keywords and the lowest for the short-length keywords. We observe also that the lowest missed detection rate occurs for long-length keywords and the highest for the short-length keywords. Finally, we observe that the task of detecting keywords belonging to the short-length keywords group will produce the higher number of false alarms in comparison with the two others groups.

In fact, by analyzing the false alarm error we observe that is usually caused by three principles causes: punctuation treating, conflict of treating word boundaries and presence of the keyword's pronunciation in other words. The results obtained using the proposed KWS system proof that more false alarms will be generated for the short-length keywords, due that they are the most probable to be a subpart of other keywords, like the long-length keywords, also they are the most probable to be composed by coupling two words on the speech. On the contrary, these effects occurs less for the long keywords. For that, we can see that the KWS system performed better on longer keywords.

From these results we can confirm that our proposed KWS system is able to localize the keywords, and yield promising performances not only in terms of accuracy but also in terms of speed. We can also confirm that the proposed deep model has shown great improvements in phone recognition decoding, representing the first stage of our proposed KWS approach, which has consequently improved the performances of our KWS system compared to our previous work using DNN, CNN, and LSTM alones.

5 Conclusion

In this paper, we presented a unified deep architecture called "Convolutional Gated Recurrent Deep Neural Network" or simply "CNN-GRU-DNN". We show that our proposed model is more competitive than all its subnetworks namely; DNN, CNN and GRU used alone. Significant improvements can be achieved, due to the complementary capabilities provided by these networks. The proposed model using two unidirectional GRU layers achieves a 0.89% relative improvement over the GRU model and 0.14% over the CLDNN model, for the TIMIT test set. And using two bidirectional GRU layers achieves a 0.72% relative improvement over the BGRU model and 0.46% over the CLDNN model. A phone error rate of 15.72% has been obtained using our proposed model with four BGRU layers and FMLLR features, which has been shown to give a very promising performance for the TIMIT phone recognition task.

These interesting results of phone recognition have ameliorated significantly the performances of our proposed two-stage keywords spotting system. Several experiments were carried out to evaluate the effectiveness of our KWS technique. And it is clear from the obtained level of performance, that we can achieve very useful and important accuracy.

As a future work, we would like to ameliorate our deep architecture by combining CNN, DNN and inside the GRU we propose to use others advanced networks, as Residual LSTM, Convolutional LSTM and Fast LSTM, etc.., in such a way we further improve the phone recognition rates and consequently improve the performances of the proposed KWS system.

References

[1] J.S.Bridle, "An efficient elastic-template method for detecting given words in running speech", British Acoustic Metting, pp.1-4,1973.

[2] A.L.Higgins and R.E.Wohlford, "keyword recognition using template concatenation", in Proceedings of the International Conference on Audio Speech and Signal Processing (ICASSP), pp. 1233-1236 (1985).

[3] J. G. Wilpon, L. R. Rabiner, C.-H. Lee, and E. R. Goldman, "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models", in IEEE Trans on Acoustics, Speech, and Signal Processing 38, 1870-1878 (1990).

[4] R.C.Rose and D.B.Paul, "A Hidden Markov Model based keyword recognition system", in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 129-132 (1990).

[5] I. Szoeke, P. Schwarz, P. Matejka, L. Burget, M. Karafiat, M. Fapso and J. Cernocky, "Comparison of Keyword Spotting Approaches for Informal Continuous Speech", in Proceedings of the 9th European Conference on Speech Communication and Technology (EUROSPEECH),pp. 633-636 (2005).

[6] D. R. H. Miller, M. Kleber, C. Kao, O. Kimball, T. Colthurst, S. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH), pp. 314-317 (2007).

[7] I. Chen, "Resource-dependent acoustic and language modeling for spoken keyword search", Ph.D. dissertation, Electrical and Compuer. Eng. Dept., Georgia Institute of technologie, 2016.

[8] J. Noguerales, "Contributions to Keyword Spotting and Spoken Term Detection For Information Retrieval in Audio Mining", Ph.D. dissertation, Eng, Inf. Dept., Madrid Univ., Madrid ESPAGNE, 2009.

[9] L. Deng and J. Platt, "Ensemble Deep Learning for Speech Recognition," in Proceedings of the 15th Annual Conference of International Speech Communication Association (INTERSPEECH), pp. 1915-1919 (2014).

[10] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to Construct Deep Recurrent Neural Networks," in Proceedings of the 2nd International Conference on Learning Representation (ICLR) (2014).

[11] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks," in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 4580-4584 (2015).

[12] R. Zazo, T. N. Sainath, G. Simko and C. Parada, "Feature learning with raw-waveform CLDNNs for Voice Activity Detection," in Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH), pp. 3668-3672 (2016).

[13] G. Dahl, D. Yu, L. Deng, and A. Acero, "Large vocabulary continuous speech recognition with context-dependent DBN-HMMs," in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (2011).

[14] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," Neural Comput., vol. 18, pp. 1527–1554, 2006.

[15] O. A. Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying Convolutional Neural Network Concepts to Hybrid NN-HMM Model for Speech Recognition," in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 4277-4280 (2012).

[16] O. A. Hamid, L. Deng, and D. Yu, "Exploring Convolutional Neural Network Structures and Optimization Techniques for Speech Recognition," in Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH) (2013).

[17] L. Tôth. "Combining time and frequency domain convolution in convolutional neural network-based phone recognition". In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 190-194 (2014).

[18] T. N. Sainath, A. Mohamed, B. Kingshury, and B. Ramabhadran, "Deep Convolutional Neural Networks for LVCSR," in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 8614-8618 (2013).

[19] D. Palaz, R. Collobert, and M. Magimai.-Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," in Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH), pp. 1766-1770 (2013).

[20] D. Palaz, R. Collobert, and M. Magimai. -Doss, "End-to-end Phoneme Sequence Recognition using Convolutional Neural Networks," ArXiv e-prints, Dec. 2013.

[21] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH) (2014).

[22] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (2014).

[23] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," Neural Net-

works, vol. 12, pp. 5–6, 2005.

[24] D. Yu, L. Deng, and G. Dahl, “Roles of pre-training and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition,” in NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2010).

[25] A. Mohamed, G. Hinton and G. Penn, “Understanding how deep belief networks perform acoustic modeling”. in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp.4273-4276 (2012).

[26] A. Mohamed, T. Sainath, G. Dahl, B. Ramabhadran, G. Hinton and M. Picheny, “Deep Belief Networks Using Discriminative Features for Phone Recognition,” in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 5060-5063 (2011).

[27] G. E. Dahl, M. Ranzato, A. Mohamed, and G. E. Hinton, “Phone recognition with the mean-covariance restricted Boltzmann machine,” in Advances in Neural Information Processing Systems 23, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, Eds., Advances in Neural Information Processing Systems, pp. 469-477 (2010).

[28] G. E. Dahl, D. Yu, L. Deng, and Al. Acero,“Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition” ,in IEEE Transactions on Audio, Speech, and Language Processing, pp. 30-42 (2012).

[29] J. Chung, C.Gulçehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in Proc. of NIPS, 2014.

[30] A. Mohamed, “Deep Neural Network acoustic models for ASR”, Ph.D. dissertation, Computer science. Dept., Toronto Univ., Toronto, U.K., 2014.

[31] N. Jaitly, P. Nguyen, AW. Senior, and V. Vanhoucken, “Application of pretrained deep neural networks to large vocabulary speech recognition”, in Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH) (2012).

[32] A.Graves, A. Mohammed, G. Hinton. “Speech recognition with deep recurrent neural networks”. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 6645-6649 (2013).

[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The Kaldi speech recognition toolkit”, in IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society, 2011.

[34] T.Dutoit, “Introduction to text-to-speech synthesis”, Kluwer academic publishers, London (1997).