Using RNN Language Model Effect on the Development of Speech Recognition System: A Review

[1]Abbas Mohamad Ali, [2*]Nawroz Ibrahim Hamadamen, [3]Haider Abdula Haddad

[1]Assistant Professor, Department of Software Engineering, Salahaddin University, Erbil - Kurdstan Region – Iraq
[2*]Lecturer, Department of Software Engineering, Salahaddin University, Erbil - Kurdstan Region – Iraq
[3]Lecturer, Department of Computer Science, Salahaddin University, Erbil - Kurdstan Region – Iraq

Email: abbas.mohamad@su.edu.krd, nawroz.hamadamen@su.edu.krd, Haider.haddad@su.edu.krd

Abstract

To be able to control gadgets by voice has continuously charmed humanity. Nowadays after seriously investigate, Speech Recognition System, have made a specialty for themselves and can be seen in numerous strolls of life. The exactness of Speech Recognition Systems remains one of the foremost imperatives investigates challenges e.g., noise, speaker changeability, dialect inconstancy, lexicon estimate and space. The plan of speech recognition system requires cautious considerations to the challenges such as different sorts of Speech Classes and Speech Representation, Speech Preprocessing stages, Include Extraction methods, Database and Execution assessment. Automatic voice recognition using recurrent neural networks (RNNs) has recently gained importance and promise on mobile devices like smart phones. However, earlier RNN compression methods either experience severe accuracy loss due to the preserved regularity for hardware friendliness or hardware performance overhead as a result of the inconsistency. RNNs are effective for simulating sequences because they have cyclic connections, opposing feedforward neural networks. For applications like handwriting recognition, language modeling, and the phonetic labeling of auditory frames, they have been successfully used for sequence labeling and sequence prediction. RNNs have only been used for small-scale tasks like phone recognition, as opposed to deep neural networks, in speech recognition. Modern speech recognition capability for comparatively small models is provided by Long Short-Term Memory (LSTM) models, which converge quickly. End-to-end voice recognition is proposed using RNN-T. In particular, Minimum Bays Risk (MBR) training is carried out by reducing the predicted edit distance between the reference label and the initialized RNN-T trained model. N-best hypothesis developed in-sequence and on the fly.The plan of speech recognition system requires cautious considerations to the challenges such as different sorts of Speech Classes and Speech Representation, Speech Preprocessing stages, Include Extraction methods, Database and Execution assessment. This paper presents the progresses made as well as highlights the squeezing issues for a speech recognition system.

Keywords: Speech Recognition, Language Model, Neural Network, Recurrent, Neural Network, Recurrent Neural Network-Transducer, Automatic Speech Recognition.

1. Introduction

A recurrent neural network (RNN) could be a sort of artificial neural network which employments consecutive information or time arrangement information. In the recent years, deep learning is emerging as other way of multilayer neural networks and back propagation preparing. Its application within the field of language model, such as limited Boltzmann machine language model, gets great results. This language

model based on neural network can evaluate the likelihood of the following word shows up agreeing to the word arrangement, which is mapped to a persistent space. This language model can illuminate the issue of sparse data. Moreover, a few researchers are developing language model making utilize of recurrent neural network mode in arrange to form full utilize of the going before content to predict the next words. From these models, we are able sort out the confinement of long-distance reliance in language. The disappearing and growing gradient issues of traditional RNNs are solved by the Long Short-Term Memory (LSTM) recurrent neural network (RNN) architecture [1]. End-to-end voice recognition is proposed using RNN-T. In particular, Minimum Bays Risk (MBR) training is carried out by reducing the predicted distance measure between the reference label sequence as well as on generated N-best hypothesis. This is initialized with an RNN-T trained model.
According to experimental results, an MBR trained model performs significantly better than an RNN-T trained model, and additional gains can be made by using an external Neural Network Language Model (NNLM) during training, [2]. Although they are currently performing worse than RNN/transformer-based models, convolutional neural networks (CNN) have demonstrated encouraging results for end-to-end voice recognition.

With a brand-new CNN-RNN-transducer architecture we call ContextNet, we bridge this gap and go beyond it. ContextNet has a deep convolution encoder that adds squeeze-and-excitation modules to convolution layers to incorporate global context information. Additionally, a straightforward scaling technique will be suggesting that grows ContextNet's widths and achieves a fair balance among both processing and correctness. By suggesting an RNN-T rescoring method to re-rank the hypotheses and using Recurrent Neural Network-Language Model RNN-LM to rescore the new N-best list [3]. The first work to successfully implement real-time RNN prediction on mobile systems is RTMobile. According to experimental findings, RTMobile can greatly beat current RNN hardware acceleration techniques in terms of inference accuracy and processing speed [4]. This paper endeavors to capture the long-distance data based on RNN. On the other hand, the energetic adjunction of language model an analyzed and outlined agreeing to the language highlights. The test result demonstrates there are impressive enhancements to the productivity of growing lexicon proceeding speech recognition utilizing RNN language model. End-to-end preparing strategies such as Connectionist Temporal Classification make it conceivable to prepare RNNs for grouping labeling problems where the input-output alignment is unknown [1]. The standard RNN more complex, but the Clockwork RNN (CW-RNN) worked on eliminate the number of RNN parameters. And it effects the performance significant and speed up the network evaluation [5].On the other hand, the problem in machine learning are sequence prediction and classification, (RNNs) have the capacity, in hypothesis, to manage with these temporal dependencies by ethicalness of the short-term memory executed by their repetitive (feedback) connections.

According to experimental results, an MBR trained model performs significantly better than an RNN-T trained model and additional gains can be made by using an external NNLM during training.

According to experimental findings, the semi-on-the-fly method can speed up the on-the-fly method by six times and produce a Word Error Rate (WER) improvement over a baseline RNN-T model of 3.6 percent. We can further improve the results if we re-rank the hypotheses using a proposed RNN-T rescoring method and execute additional rescoring using an external RNN-LM. The top system obtains an 11.6 percent WER reduction on music-domain utterances and a 5 percent relative improvement on an English test-set of real far-field recordings. Recurrent neural network transducer (RNN-T), attention-based seq2seq models, and connectionist temporal classification (CTC) based models are a few examples of such models. The best streaming end-to-end recognizer among these models is RNN-T, which has demonstrated competitive performance when compared to traditional systems. RNN-T loss, which tries to increase the log-likelihood of training data, is generally used to train RNN-T models. However, only a small amount of research has examined sequential discriminative training standards for RNN-T models. A state-level minimum Bayes risk (sMBR) training criterion has been effectively utilized for traditional hybrid systems. In order to reduce expected WER for word-level MBR training, sampling-based methods for CTC and recurrent neural aligner (RNA) were developed. Attention-based seq2seq models may now be trained with considerable gains thanks

to minimum WER (MWER) training. Recent research in suggested using the decoded alignments of N-best hypotheses to perform MBR training in the context of RNN-T models.

For RNN-T, a brand-new and effective MWER training strategy is suggested. We recalculate the scores of all possible alignments for each hypothesis in a given N-best list and use the combined scores for MWER training in contrast to the existing method in, which uses relatively small beam size to perform on-the-fly decoding to generate alignments scores and N-best list.

The forward-backward approach, which is similar to RNNT training in terms of speed and memory utilization, is used to compute the hypothesis probability scores and back-propagation gradients. Decoding and MWER training for each subset can be executed offline iteratively, allowing us to accelerate both the decoding and training processes separately because on-the-fly N-best creation is expensive and because the N-best lists don't change significantly during a short training time. We demonstrate that the semi-on-the-fly decoding and training method may speed up the MWER training process by 6 times without harming WER improvement by using the proposed methodologies on large-scale far-field English data sets (3.6 percent) [6].

This paper endeavors to capture the long-distance data based on RNN. On the other hand, the energetic adjunction of language model an analyzed and outlined agreeing to the language highlights. The test result demonstrates there are impressive enhancements to the productivity of growing lexicon proceeding speech recognition utilizing RNN language model. End-to-end preparing strategies such as Connectionist Temporal Classification make it conceivable to prepare RNNs for grouping labeling problems where the input-output alignment is unknown [1]. The standard RNN more complex, but the Clockwork RNN (CW-RNN) worked on eliminate the number of RNN parameters. And it effects the performance significant and speed up the network evaluation [5]. On the other hand, the problem in machine learning are sequence prediction and classification, (RNNs) have the capacity, in hypothesis, to manage with these temporal dependencies by ethicalness of the short-term memory executed by their repetitive (feedback) connections.

2. Literature Review

RNN execution in speech recognition has so distant been baffling, with much better results returned by deep feedforward networks. In speech recognition [1] neural network have a long history in mixed with hidden Markov models. It is conceivable to prepare RNNs ‘end-to-end’ for speech recognition, rather than combining RNNs with HMMs. Another founded result is expanding the system to range vocabulary speech recognition. When the long-term memory is required, train successfully is difficult [5] proposed simple system and changed in the structure of RNN; CW-RNN worked on the hidden layer and portions into different modules. However, does not cover classes of problems, like such reinforcement learning, the bigger set of connectionist models for successive information. Tests appear that [7] compared with the standard crossover DNN frameworks, Eesen accomplishes comparable word mistake rates (WERs), whereas at the same time speeding up interpreting essentially.

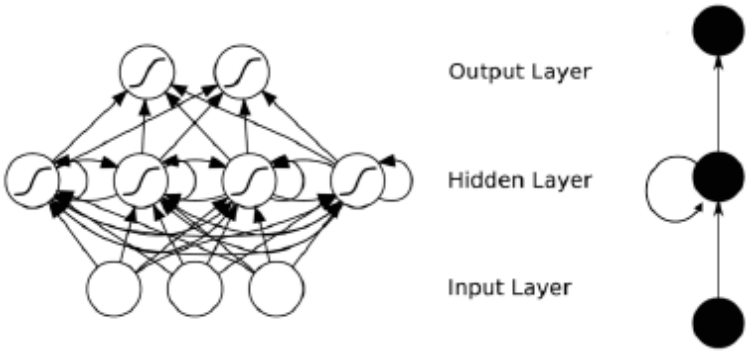


Figure 1 Classic Form of RNNs

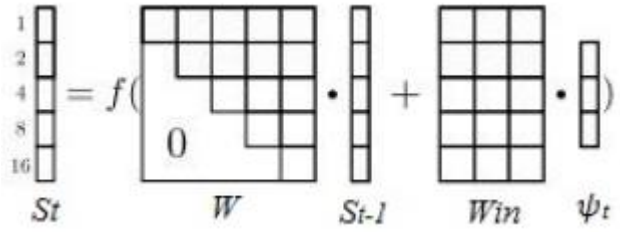


Figure 2 Input and Output Connection in CW-RNN, with 5 Hidden Layer Groups at step t = 6

Input and Output Connection in CW-RNN with 5 Hidden Layer Groups at step t = 6 [8]. However, the proposed system cannot cover due to the evacuation of GMMs, acoustic modeling in ease cannot use speaker-adjusted front-ends. [9] show procedures that advance make strides execution of LSTM RNN acoustic models for huge lexicon speech recognition. It was produced the utilize of longer term feature representations, prepared at lower frame rates brought stability to the joining of CTC training of models.[10] altogether progressed current best end-to-end ASR framework without any linguistic assets such as morphological analyzer and pronunciation dictionary, which are basic components of customary Mandarin Chinese and Japanese ASR frameworks. [11] Due to the difficulty of modeling linguistic restrictions across long sequences of characters, character-based LMs often underperform word LMs for languages with a phonogram alphabet utilizing fewer different characters, such as English. In terms of character sequence length, English sentences are substantially longer than Japanese and Chinese ones. In comparison to the word-based LM, the character-based LM provides the following advantages in the decoding process:

1. Character-based LM can aid in the survival of right hypotheses. During the beam search, they are rescored at word boundaries. The identification of the hypothesis is established before it reaches the boundary.

The last word is unknown, and its likelihood cannot be calculated. As a result, accurate character-level prediction is critical. to avoid pruning errors when there are multiple hypotheses in a single word [12], Character-based LM can predict character sequences in even the most difficult cases. OOV words that are not in the word-based vocabulary LM, because the word-based LM cannot forecast the unknown sequences of characters excellent. [10] have proposed a multi-level LM, in which word-based and character-based RNNLMs are combined in half-breed CTC/attention-based ASR. And it works as shown in figure 3.

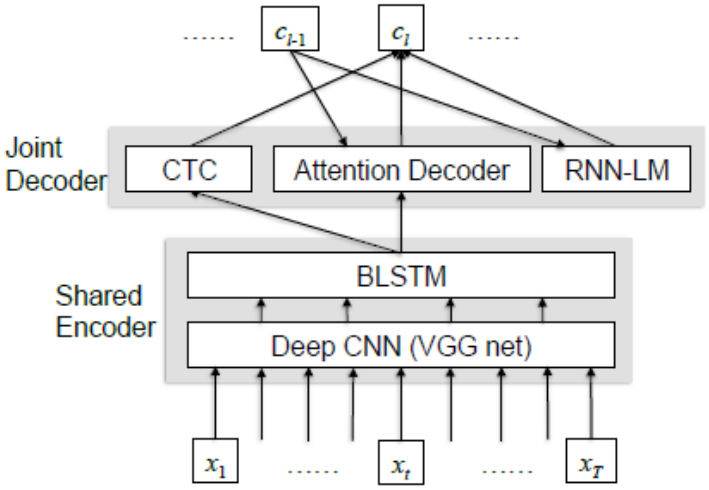


Figure 3 A VGG net is followed by BLSTM layers and an LM extension in a hybrid attention/CTC network with LM extension: the shared encoder has a VGG net, followed by BLSTM layers and an LM extension. At the same time, both CTC and attention model objectives are being trained. The CTC, attention, anticipates an output label sequence via the joint decoder.

RNN-T models are commonly trained with RNN-T loss, which tries to increase the training data's log-likelihood. However, only a small number of studies have looked into sequential discriminative training criteria for RNN-T models. A state-level minimum Bayes risk (sMBR) training criteria has been effectively applied to traditional hybrid systems [6].

The encoder and prediction networks of an RNN-T model frequently have different model structures, making it difficult to train them at the same time. Directly training RNN-T from random initialization may result in a model that is biased toward one of the model components, such as audio or linguistic input. The majority of organizations use an initialization technique that uses a CTC model for the encoder and an RNNLM for the prediction network. CTC, on the other hand, produces a succession of spikes separated by a blank. Because of the CTC-based pre-training, most encoder output hence t leads to blank, resulting in incorrect inference for the RNN-T model.

To pertain the encoder with the Cross Entropy (CE) criterion, use external alignments, rather than a CTC model. The encoder is seen as a token classification model.

The CE loss is used to train an RNN-based token classification model, as illustrated in the right side of Figure 4. Use the terms' CE losses and 'CTC losses to refer to the cross entropy loss function and the CTC forward-backward algorithm-based loss.

'RNN-T loss' is used to symbolize the RNN-T loss function. We can determine the boundary frame index of each word using word-level alignments. We split the total frames inside the word border evenly among the word pieces when the word is divided into more than one word piece. We can determine the boundary frame index of each word using word-level alignments. We allot so same amount of time to a word that is divided into more than one word fragment.

The total number of frames inside the word boundaries divided by the number of word components. There will be a rare instance where a term contains multiple word components. We cannot generate token alignments since we are using frames rather than frames. This specific case's overall ratio is less than 0.01 percent of all training.

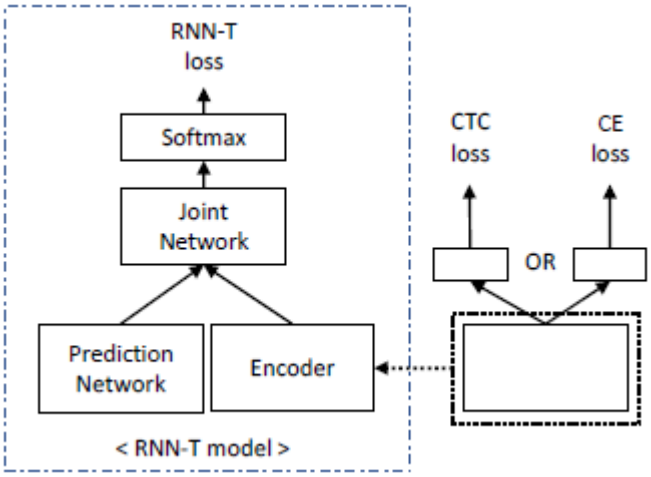


Figure 4 The dashed arrow in this indicates Initializing from a pre-trained model illustration of encoder pre-training for RNN-T.

On top of the encoder, one more completely linked layer is created based on the encoder structure, with the output h_t^{enc} used for token classification as in equation (1).

$$L_{enc} = -\sum_{k=1}^{K} Y_{t,k} * Log(softmax(f^{fc}(h_{t,k}^{enc}))) \dots\dots\dots\dots\dots (1)$$

where f^{fc} stands for fully connected layer, k stands for label index, and K stands for target dimension, which is also the dimension of $z_{t;u}$. For each input frame x_t, y_t is the word piece label.

Following the encoder pre-training, each output h_t^{enc}, which is a high-level representation of input acoustic features, should have alignment information. An example of the whole-network pre-training is illustrated in figure 5.

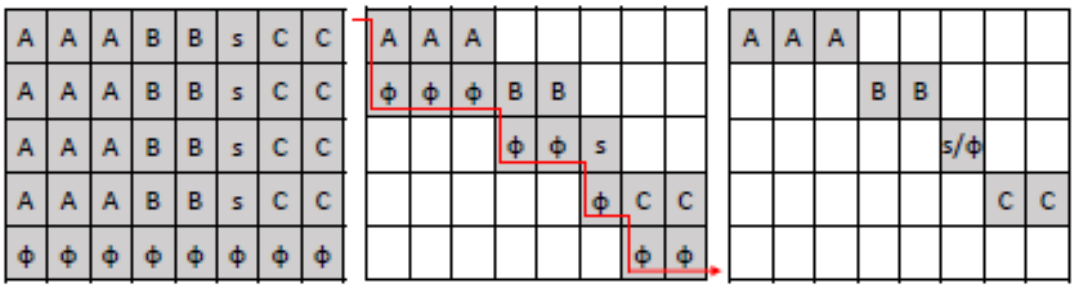


Figure 5 Three label tensors that have been constructed for whole-network pre-training. Each grid corresponds to a single-hot vector. 'A B s C' is an example 8-frame utterance, with the alignment 'A A A B B s C C'. The letters's' and "stand for space and blank, respectively, in each label tensor. Each sub-figure reflects the order y1 to y3 from left to right.

For CE computing, only gray grids are used. When decoding, the red arrow in y2 reflects the decoding path [13]. The goal of speaker adaptation techniques is to improve the performance of a voice recognition system for a particular speaker or set of speakers. It can be achieved by either changing a pre-trained speaker into a self-contained speaker. Altering the (SI) model to match the target speaker, the target speaker's features to match the SI system's pre-trained features, the target speaker's adaption data.

Many speaker adaption approaches have recently been presented and demonstrated to be effective in hybrid NN/HMM. Linear Input Network ((LIN), Linear Hidden Network (LHN), and Linear Output Network (LON) are all examples of SI neural networks that aim to add additional transforming layers.

Estimate the adaptation parameters using maximum a posteriori (MAP) linear regression, which naturally incorporates prior knowledge into the adaptation process, to improve the robustness of adaptation. Recently several quick speaker adaptation methods based on the so-called speaker code have been presented for DNN and CNN, and have demonstrated to be a promising adaptation approach in speaker adaptation.

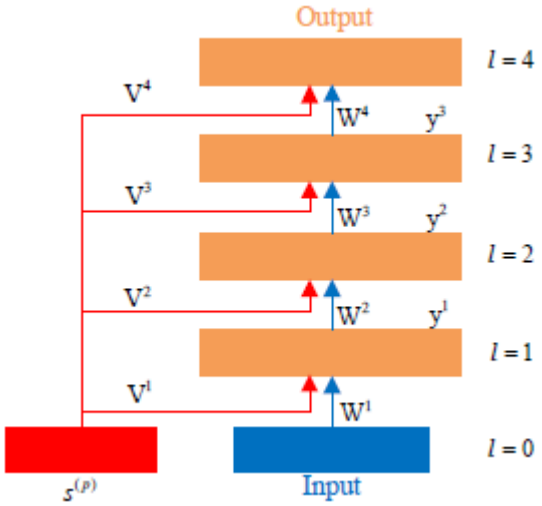


Figure 6 Speaker code-based adaptation model structure (n = 5 Levels)

The structure of the speaker code-based adaption model suggested in this model can be regarded as a common model, as shown in Figure 6.

Distinct neural network (NN) models, such as Deep Neural Network DNN and recurrent neural network with bidirectional Long Short-Term Memory RNN-BLSTM, have different adaption structures.

Figure 7 shows an LSTM memory cell with one self-connected cell and three regulating gates (time-delayed connections are indicated by red lines). The input and output gates in the memory cell control the flow of data into and out of the cell.

In the meanwhile, the forget gate is utilized to allow the cell to reset itself. Additionally, peephole weights connect the gates to the cell, which are used to get more precise Constant Error Carousel (CEC) data.

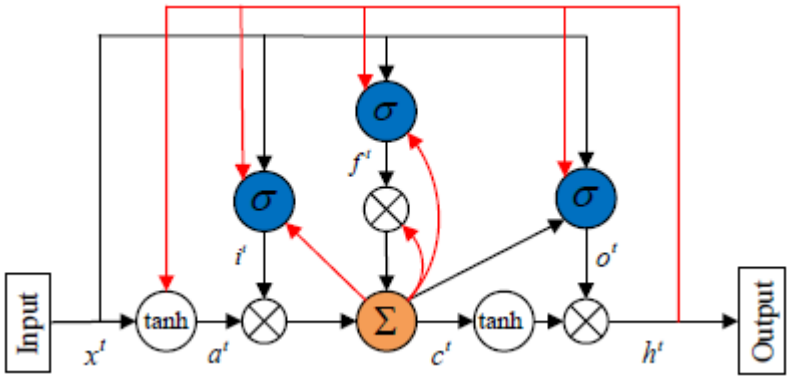


Figure 7 The LSTM network architecture includes a memory cell.

Table 1 shows that when given a speaker code size of 500 to 2000, adaptation performance is not very sensitive to it (PER ranging from 18.8% to 19.3%).

In PER, however, a speaker code size of 300 performs 19.9% better (with a 4.78 percent reduction in relative phone error). This is most likely due to the 300-character speaker code being too small to model the information of the target speaker. Furthermore, with a speaker code size of 1500, SA-CIAF produces the greatest results (with a relative phone mistake reduction of 10.05 percent) [14].

Table 1: SA-CIAF PERs (in percent) on RNN-BLSTM (3*250) with various speaker code sizes.

SC size	baseline	SA-CIAF
300	20.9	19.9
500		19.3
1000		19.2
1500		18.8
2000		19.0

End-to-End (E2E) systems is an emerging subject in automatic speech recognition research. The most popular three are the Attention Encoder-Decoder (AED), RNN Transducer (RNN-T), Connectionist Temporal Classification (CTC).

RNN-T has the advantage of being able to do online streaming, which is difficult for AED, and it does not use CTC's frame-independence assumption [15].

Automatic speech recognition (ASR) is a well-established collection of technologies that enables successful user interface applications like voice search. Current systems, on the other hand, rely largely on the scaffolding of complex legacy architectures based on classical methodologies, such as hidden Markov models (HMMs), Gaussian mixture models (GMMs), hybrid HMM/deep neural network (DNN) systems, and sequence discriminative training methods [11]. For languages lacking apparent word borders, these systems also require hand-made pronunciation dictionaries based on linguistic assumptions, additional training procedures to build context-dependent phonetic models, and text preprocessing such as tokenization.

As a result, developing ASR systems for new applications, especially for new languages, is difficult for non-experts.

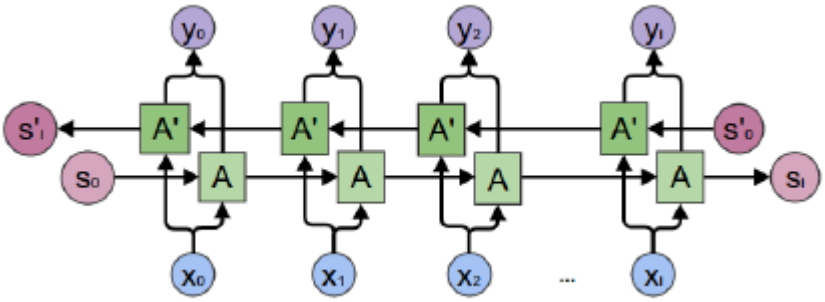


Figure 8 LSTM Formation

The word-based RNN-LMs using a huge Libri-Speech, corpus it was effective. It explores an optimization process [16] for (RNN) based SAD and compared three types of RNNs such as basic RNN, long short-term memory (LSTM) network with peepholes, and a coordinated-gate LSTM (CG- LSTM). [17] investigate RNN-T for a Chinese expansive lexicon ceaseless speech recognition (LVCSR) task and point to disentangle the preparing handle whereas keeping up performance. First, a modern methodology of learning rate rot is proposed to accelerate the show meeting. Moment, we discover that including convolutional layers at the starting of the network and utilizing ordered data can dispose of the pre-training handle of the encoder without loss of execution. [18] experimentally compared and analyzed Transformer and customary recurrent neural systems (RNN) in add up to of 15 ASR, one multilingual ASR, one ST, and two TTS benchmarks. Confirmed [19] that the between three methods Connectionist Temporal Classification (CTC), Attention Encoder-Decoder (AED), and RNN Transducer (RNN-T) the RNN-T is better than of them. It is worked on improving the RNN-T training in two related fields reduce the memory consumption and propose better model structures to good accuracy but small footprint. Demonstrated the achieve up-to 11.8% relative word error rate (WER). Suppose [2] the minimum Bayes risk (MBR) training of RNN-T for end-to-end speech recognition; MBR prepared system achieves outright character error rate (CER) reductions of 1.2% and 0.5% on examined and unconstrained Mandarin speech individually over a solid convolution and transformer based RNN-T pattern prepared on 21,000 hours of speech. () assess RNN, Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) to compare their performances on a decreased TED-LIUM speech data set.

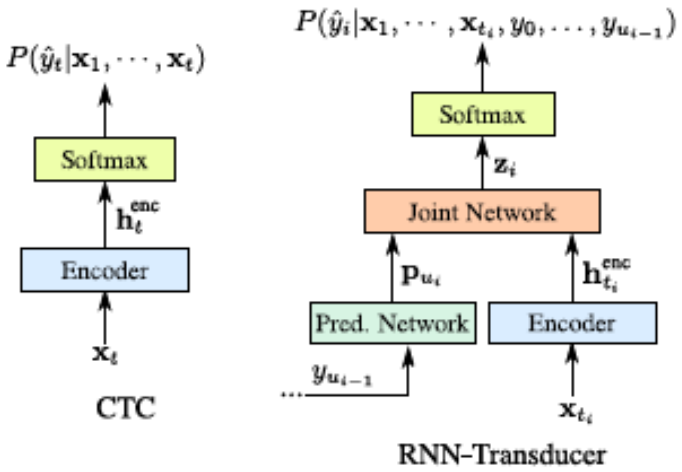


Figure 9 Diagram illustrative of CTC and RNN-T

The finds illustrate that LSTM accomplishes the best word error rates; in any case, the GRU optimization is speedier whereas achieving word error rates close to LSTM. However, does not cover the learning rate, dropout rate as well as higher numbers of neurons in the hidden layers.

The RNN-T model, which comprises of encoder, prediction, and joint networks, is depicted in Figure 9. The encoder network is similar to the acoustic model, which transforms acoustic feature x_t into a high-level representation h_t^{enc}, where t is the time index.

$$h_t^{enc} = f^{enc}(x_t) \dots\dots\dots\dots\dots (2)$$

The prediction network operates in the same way as an RNN language model, producing a high-level representation h_u^{pre} by conditioning on the previous non-blank target y_{u-1} predicted by the RNN-T model, where u is the output label index.

$$h_u^{pre} = f^{pre}(y_{u-1}) \dots\dots\dots\dots\dots (3)$$

The joint network is a feed-forward network that combines the encoder and prediction network outputs as h_t^{enc} and h_u^{pre}.

$$z_{t,u} = f^{joint}\left(h_t^{enc}, h_u^{pre}\right) \dots\dots\dots\dots\dots (4)$$

$$= \psi\left(Uh_t^{enc} + Vh_u^{pre} + b_z\right) \dots\dots\dots\dots\dots (5)$$

U and V are weight matrices, b_z is a bias vector, and is a nonlinear function, such as Tanh or ReLU.

A linear transform connects the $z_{t,u}$ to the output layer.

$$h_{t,u} = W_{yz_{t,u}} + b_y \dots\dots\dots\dots\dots (6)$$

After applying the SoftMax procedure, the final posterior for each output token k is produced.

$$P(k|t,u) = softmax\left(h_{t,u}^k\right) \dots\dots\dots\dots\dots (7)$$

The negative log posterior of output label sequence y given input acoustic feature x is the RNN-T loss function [15].

$$L = -\ln P(y|x) \dots\dots\dots\dots\dots (8)$$

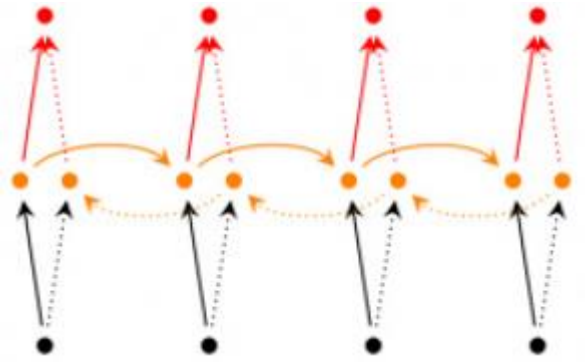


Figure 10 Bidirectional RNNs Formation.

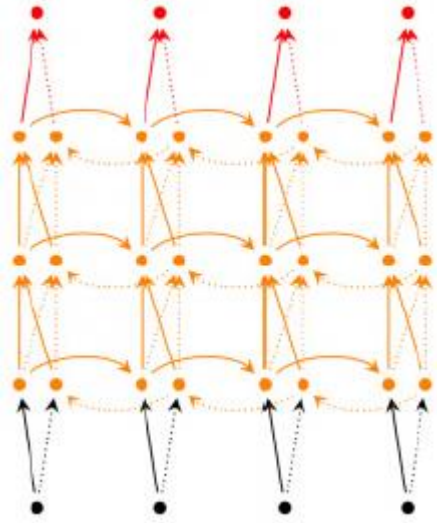


Figure 11 Deep RNNs Formation.

Due to the deep neural network (DNN) high prediction accuracy in many artificial intelligence applications, such as image identification, it has become the state-of-the-art approach. Speech recognition, characterization, and recommender system. Recurrent neural networks (RNNs) are one type of DNN architecture. For speech recognition, neural networks (RNNs) are commonly utilized, because they contain cycles for transferring information. When reading inputs, neurons are involved. Gated Recurrent, for example, is a type of gated recurrent network. The most recent representative type of RNNs is the GRU [4]. It has a lot of success with automatic voice recognition. In terms of compression rate, inference accuracy, execution time, and energy economy, experimental results show that RTMobile greatly surpasses existing RNN hardware acceleration approaches [4].

Block-based structural pruning and compiler-assisted speed optimization are the two major components of RTMobile. Our innovative block-based structured pruning methodology, unlike existing structured pruning

approaches used on DNNs, may give a finer pruning granularity to retain excellent inference accuracy while drastically lowering RNN model size. On mobiles, we also suggest many compiler-based optimization strategies for determining block size and generating the best code.

1. The Gated Recurrent Unit (GRU) is a gated recurrent unit that merges the forget and input gates into a single "update gate." It also modifies the cell state and hides the state, among other things.

The resulting GRU model is easier to understand than ordinary LSTM models, and it is gaining popularity. Figure 12 depicts a single GRU whose functionality is developed iteratively from t = 1 to T using the following equations. The update gate, output gate, cell state, and cell output are represented by the symbols z, r, and h, respectively. GRU is a more advanced RNN than LSTM since it is a more advanced type of RNN.

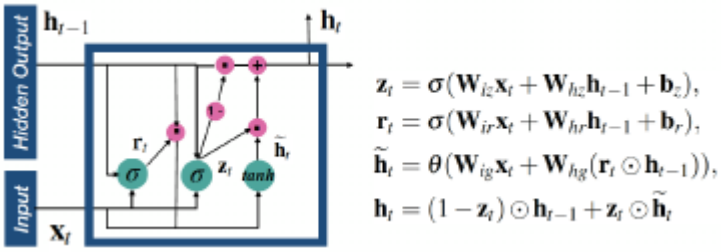


Figure 12 There is just one GRU model.

1. Techniques for Compressing DNN Models, DNN weight pruning, as a representative technique in DNN model compression, removes duplicated or less significant weights

to reduce inference phase storage and processing costs. Weight pruning is divided into two types: non-structured pruning and structured pruning.

2. On Mobile Devices, DNN Acceleration, MCDNN, DeepMon, TFLite, TVM, and Alibaba

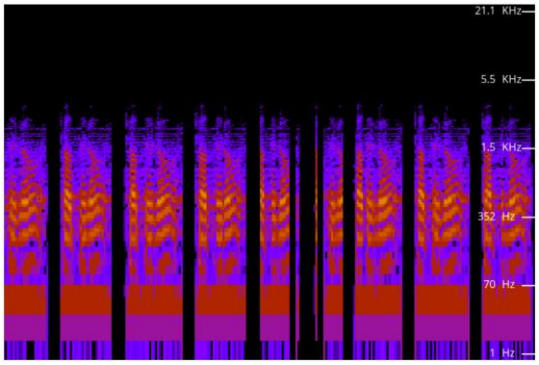
Mobile Neural Network are only a few of the recent attempts aimed at speeding up DNN execution on mobile devices. However, unlike RTMobile, most of them do not fully utilize model compression techniques. None of the available frameworks, in particular, can provide RNN acceleration on mobile devices [4].

The capacity of machines to recognize speech automatically (ASR) minimizes the complexity of communication between humans and machines. The most common method of communicating with machines has been through written instructions.

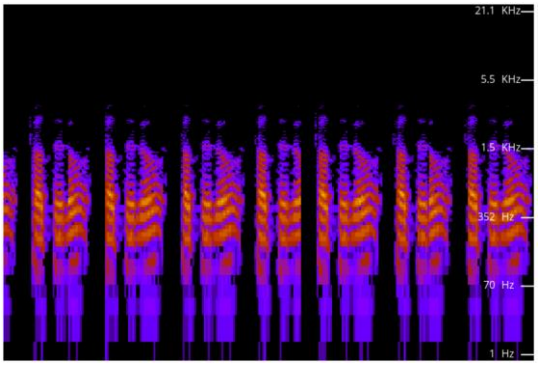
Signal Analysis: Extracting Features MFCC

The unprocessed audio speeches are unsuitable for direct usage. The signals contain a lot of duplicated information as well as noise, which can make detection difficult.

The spectra of the identical audio source before and after noise reduction are shown in Figure 13. As a result, only the most distinguishing aspects of the signals must be retrieved.



Before noise reduction, the spectrum



After noise reduction, the spectrum

Figure 13 Noise in audio signals has an effect

A two-layer deep recurrent neural network with 100 LSTM cells each Our model is shown in Figure 14. The input layer is the bottommost layer, where each time frame of a specific example is injected at each time step. The time frames' coefficients are stored in 13 units in the layer. The LSTM recurrent layers are the next two layers. As it has been detected 30 distinct phonemes, the last layer is a SoftMax output layer with 30 units. When tiny batches of instances are input into the network, a probability distribution over each phone is generated, and the phone with the highest probability is chosen.

Modeling sequences is particularly efficient using the coupled LSTM cells. Figure 15 depicts an LSTM cell [20].

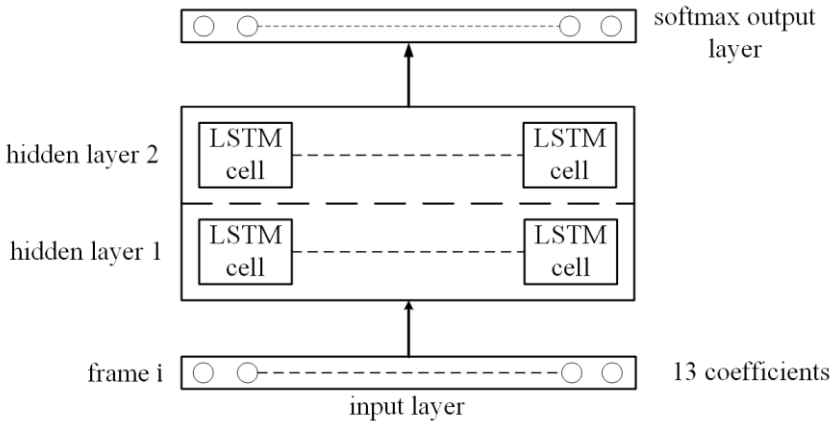


Figure 14 Speech recognition using the LSTM recurrent neural network

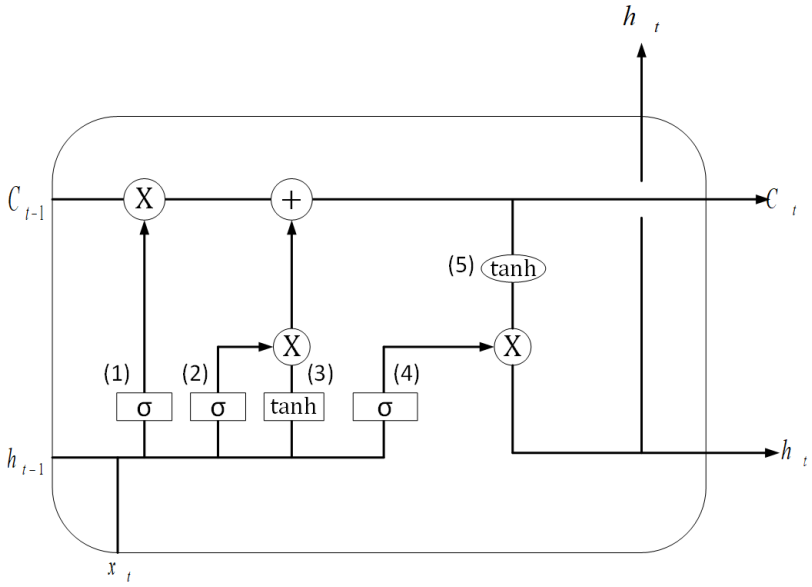


Figure 15 Shows an LSTM cell.

The model comprises of an 'encoder,' which is partially initialized from a recurrent neural network language model trained on text input alone, and a 'decoder,' which is partially initialized from a connectionist temporal classification-based (CTC) audio model. The RNN-T loss is used to train the complete neural network, which then outputs the recognized transcript as a sequence of graphemes, allowing for end-to-end speech recognition. We discovered that using sub-word units ('wordpieces') to capture longer context and greatly minimize substitution errors can boost performance even further. The best RNN-T system, which consists of a twelve-layer LSTM encoder and a two-layer LSTM decoder trained with 30,000 wordpieces as output targets, achieves a word error rate of 8.5 percent on voice-search and 5.2 percent on voice-dictation tasks, which is comparable to a state-of-the-art baseline of 8.3 percent on voice-search and 5.4 percent on voice-dictation.

Word counts from text data are used to train a statistical wordpiece model for segmenting each word into subwords. In subword units, there is an additional space symbol. tor> to> ise> space> and> space> the> hare> re> is an example segmentation for the sentence tortoise and the hare.

Wordpieces provide a better balance than graphemes, with more context and a variable amount of labels. More common words show as a single label since the wordpiece model is based on word frequencies. Terms like 'mall, remember', and 'doctor' appear in a vocabulary of 1,000 generated wordpieces, while less common words like 'multimedia,' 'tungsten,' and '49er' appear in a vocabulary of 30,000 created wordpieces. We explored with deep LSTM networks for the encoder networks in RNN-T models (5 to 12 layers). We employed a two-layer LSTM network, a feed-forward layer, and a SoftMax layer for the decoder networks. We looked at different ways of initializing encoder and decoder network parameters from pre-trained models, in addition to training models using random parameter initialization. It has been previously demonstrated that for the phoneme identification task, initializing RNN-T encoder parameters from a model trained with the CTC loss is helpful. It tried initializing encoder networks from CTC loss models and

initializing LSTM layer parameters in prediction networks from LSTM language models trained on text data. The whole RNN-T model weights are trained with the RNN-T objective after initialization of encoder and prediction network weights from separate pre-trained models. Figure 16 depicts one example architecture for the RNN-T wordpiece model. The pre-trained CTC LSTM acoustic model and LSTM language model architectures used to initialize the encoder and prediction network weights are also shown in the picture. The pre-trained layers used to initialize certain layers in the RNN-T model, are indicated by dotted arrows. The CTC loss is used to pre-train the encoder networks in RNN-T models, which use phonemes, graphemes, and wordpieces as output units. We look at encoder architectures with multitask training employing hierarchical-CTC and different 'hierarchies' of CTC losses at different depths in the encoder network [21].

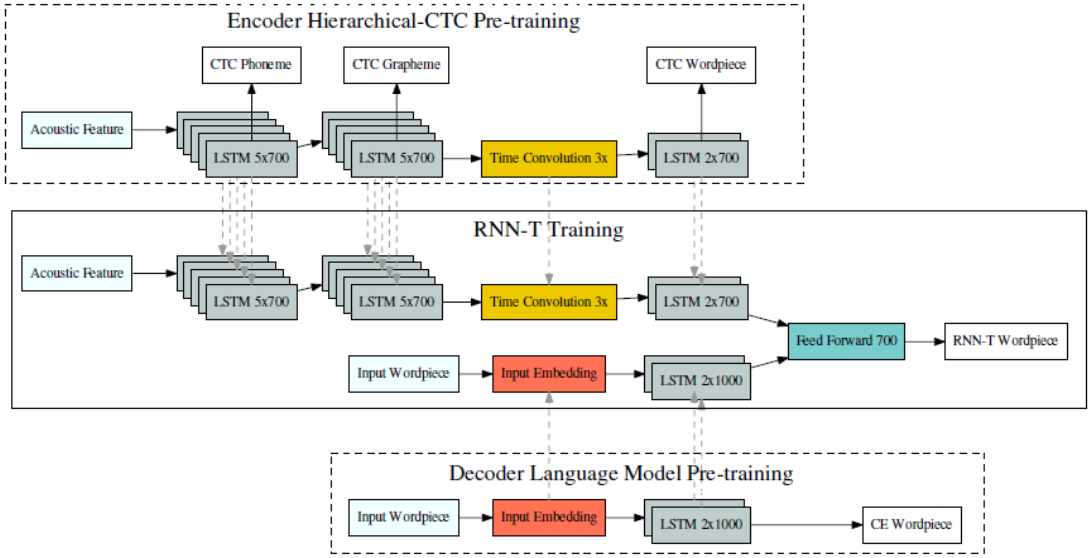


Figure 16 depicts the various steps of wordpiece RNN-T training. At 5, 10, and 12 LSTM layers, the encoder network is pre-trained as a hierarchical-CTC network that predicts phonemes, graphemes, and word pieces simultaneously. The length of the encoder time sequence is reduced by a factor of three when using a time convolutional layer. . The decoder networks is trained as an LSTM language model that predicts word pieces using cross-entropy loss optimization. Finally, the two pre-trained models are used to initialize the RNN-T network weights, as indicated by the dotted lines, and the entire network is optimized using the RNN-T loss.

End-to-end methods resulted in a remarkable reduction of both training and decoding pipelines when compared to traditional approaches, which incorporate diverse knowledge sources in a complex search algorithm. This resulted in a fast-changing research landscape in end-to-end modeling for ASR, with the most prominent examples being Recurrent Neural Network Transducers (RNN-T) and attention-based models. RNN-Ts are a perfect match for the left-to-right nature of speech, whereas attention-based models thrive at non-monotonic alignment challenges like translation.

Despite, or perhaps because of, the substantially simpler implementations, end-to-end models can now achieve unparalleled levels of voice recognition performance.

End-to-end models have been demonstrated to clearly outperform traditional approaches when given enough training data. Nonetheless, data sparsity and overfitting are intrinsic concerns in any direct sequence-to-sequence model, and numerous ways to incorporate meaningful variations and alleviate these issues have been proposed. On one public corpus (English conversational telephone voice 300 hours) and two internal datasets, the effectiveness of the proposed approaches was investigated (Spanish and Italian conversational speech 780 hours and 900 hours, respectively) [22].

Frame-level alignments between au-dio and output symbols are not required for end-to-end training of recurrent neural network transducers (RNN-Ts). As a result, the posterior lattices generated by the prediction distributions from multiple RNN-Ts trained on the same data can vary significantly, posing additional issues in knowledge distillation between such models. The differences between an offline and a streaming model are most noticeable in the posterior lattices, which is to be expected given that the streaming RNN-T emits symbols later than the offline RNN-T. We can train an offline RNN-T that can serve as a good teacher for a student streaming RNN-T using this strategy. Experiments on the standard Switchboard conversational telephone voice corpus show that knowledge distillation from an offline bi-directional counterpart improves accuracy for a streaming uni-directional RNN-T. From an offline RNN-T with a bidirectional encoder network (bi-directional RNN-T) to a streaming RNN-T with a uni-directional encoder network, use knowledge distillation (unidirectional RNN-T). Frame-level forced alignments between auditory features and output symbols are used to train traditional DNN/HMM hybrid models. Thus, by minimizing Kullback-Leibler (KL) divergence between posterior distributions from instructor and student models at corresponding frames, naive knowledge distillation worked successfully. E2E models, unlike hybrid models, are often trained from pairs of acoustic characteristics and output symbols without frame-level alignments, posing a new set of obstacles in knowledge distillation between them. We aligned posterior peaks3 for acoustic features at each time step from multiple CTC models to achieve knowledge distillation between CTC models. Because posterior distributions in RNN-Ts are conditioned not only on acoustic features but also on output symbols predicted in the past, calculating posterior distributions for acoustic features at each time step without taking into account past symbols is not straightforward. The combined network is commonly implemented as a sum of linear transformations of both embedding.

$$Z_{t,u} = \psi\left(W^{enc} + W^{pred} h_u^{pred} + b\right) \dots\dots\dots\dots\dots (9)$$

where W_{enc} and W_{pred} are weight matrices, and b is a bias, and ψ is hyperbolic tangent. To calculate a posterior distribution, zt;u is subjected to another linear transformation followed by a SoftMax operation $P(\hat{y}_t + u|t,u)$ over the set $y \cup \{\emptyset\}$. As a result, P(y t+u|t,u) defines a posterior lattice, as illustrated in Figure 17, with each P(yt+ujt; u) defining a posterior lattice. As shown in Figure 17, a posterior lattice is defined, with each node representing the posterior distribution. RNN-T training is achieved using these definitions by minimizing the RNN-T loss LRNN-T, which may be efficiently computed using a forward backward method [12].

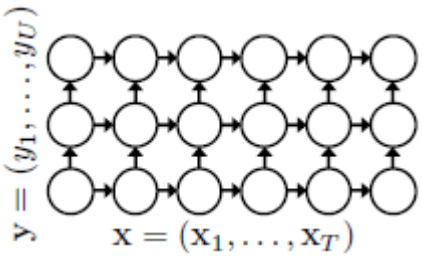


Figure 17 RNN Transducer Posterior Lattice

Conclusion

In this paper, review and assessed speech recognition system based on RNN language model, evaluate RNN, RNN-T, CTC, and compared their performances on a reduced RNN train speech data set. Many researchers' studies at the field of speech recognition system on character-based and word-based. However, according to the results achieved in the previous study the performance of RNN is good and accurate.

References

1. Alex Graves, "Generating Sequences with Recurrent Neural Networks", 2014, arXiv:1308.0850v5[cs.NE].
2. Chao Weng, Chengzhu Yu, Jia Cui, Chunlei Zhang, Dong Yu Tencent AI Lab, Bellevue, USA cweng@tencent.com. "Minimum Bayes Risk Training ofRNN-Transducer for End-to-End Speech Recognition", 2019.
3. Wei Han, Zhengdong, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, Yonghui Wu, "Context Net: Improving Conventional Neural Networks for Automatic Speech Recognition with Global Context", 2020
4. Peiyan Dong, Siyue Wang, Wei Niu, Chengming Zhang, "RTMobile: Beyond Real-Time Mobile Acceleration of RNNs for Speech Recognition", © 2020, IEEE
5. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun," Faster R-CNN: Towards Real- Time Object Detection with Region Proposal Networks", 2016, arXiv:1506.01497v3 [csCV].
6. Jinxi Guo, Gautam Tiwari, Jasha Droppo, Maarten Van Segbroeck, Che-Wei Huang, Andreas Stolcke, Roland Maas, "Efficient Minimum Word Error Rate Training of Transducer for End-to-End Speech Recognition", Amazon.com, 2020, USA.
7. Yajie, M., Mohammad, G., Florian, M. EESEN: End-to-End Speech Recognition using Deep RNN Models and WFST-Based Decoding. 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 167-174, 2015
8. Xiaoping Jiang, Jing Sun, Member, IEEE, Chenghua Li, and Hao Ding, "Video Image De-fogging Recognition based on Recurrent Neural Network", © 2018 IEEE.
9. Hasim S., Andrew S., Kanishka R., Francoise B., "Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition", ArXiv, abs/1507.0694. 2015
10. Hori, T., Watanabe, S., Zhang, Y., & Chan, W., 2018, "Advances in Joint CTC-Attention Based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM INTERSPEECH".
11. Takaaki Hori, Jaejin Cho, Shinji Watanabe, "END-TOEND SPEECH RECOGNITIO WITH WORD-BASED RNN LANGUAGE MODELS", ©2018 IEEE.
12. Gakuto Kurata, George Saon, and IBM Research, " Knowledge Distillation from Offline to Streaming RNN Transducer for End-to-end Speech Recognition", 10.21437/Interspeech.2020.

13. Hu Hu, Rui Zhao, Jinyu Li, Liang Lu, Yifan Gong, "EXPLORING PRE-TRAININ WITH LIGNMENTS FOR RNN TRANSDUCER BASED END-TO-END SPEECH RECOGNITION" ©2020 IEEE.
14. Zhiying Huang, Hao Li, Ming Lei, "DEVICETTS: A SMALL-FOOTPRINT, FAST, STABLE NETWORK FOR ON-DEVICE TEXT-TO-SPEECH", 2021.
15. Jinyu Li, Rui Zhao, Hu Hu∗ , and Yifan Gong, "IMPROVING RNN TRANSDUCER MODELING FOR END-TO-END SPEECH RECOGNITION", 2019, arXiv:1909. 1241.v1[cs.CL].
16. Gregory Gelly and J. Gauvain, "Optimization of RNN-Based Speech Activity Detection," in IEEE/ACM Transactions on Audio, Speech, and Language Processing.vol.26,no.3,pp.646-656,March 2018 doi: 10.1109/TASLP.2017.2769220
17. S. Wang, P. Zhou, W. Chen, J. Jia, and L. Xie, "Exploring RNN-Transducer for Chinese Speech Recognition", 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 2019, pp.1364-1369, doi:10.1109/APSIPAASC47483.2019.9023133.
18. Shigeki Karita, Yotaro Kubo, Michiel Adriaan Unico Bacchiani, Llion Jones, "A Comparative Study on Neural Architectures and Training Methods for Japanese Speech Recognition", 2021, rXiv: 2106.05111v1 [cs.CL].
19. Yanzhang He, Tara N. Sainath_, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, Qiao Liang, Deepti Bhatia, Yuan Shangguan, Bo Li, Golan Pundak, Khe Chai Sim, Tom Bagby, Shuo-yiin Chang, Kanishka Rao, Alexander Gruenstein, "STREAMING END-TO-END SPEECH RECOGNITION FOR MOBILE DEVICES", ©2019 IEEE.
20. Md Mahadi Hasan Nahid, Bishwajit Purkaystha, & Md Saiful Islam, "Bengali Speech Recognition: A Double Layered LSTM-RNN Approach", 2017 IEEE.
21. Kanishka Rao, Hasim Sak, Rohit Prabhavalkar, "Architectures Data and Units Exploring For Streaming End-To-End Speech Recognition With Rnn-Transducer", ©2017 IEEE
22. George Saon, Zolt´an T¨uske, Daniel Bolanos and Brian Kingsbury, "Advancing Rnn Transducer Technology For Speech Recognition", ©2021 IEEE.
23. G. B. Loganathan, T. H. Fatah, E. T. Yasin and N. I. Hamadamen, "To Develop Multi-Object Detection and Recognition Using Improved GP-FRCNN Method," *2022 8th International Conference on Smart Structures and Systems (ICSSS)*, 2022, pp. 1-7, doi: 10.1109/ICSSS54381.2022.9782296.
24. Babu Loganathan, Ganesh (2021) *Recent Scope for AI in the Food Production Industry Leading to the Fourth Industrial Revolution.* Webology, 18 (2). pp. 1066-1080.
25. Ganesh Babu Loganathan, Nawroz Ibrahim Hamadamen, Elham Tahsin Yasin, Amani Tahsin Yasin, Alaa Amer Mohammad, Israa Nabeel Adil, Sidra Bahjat Ismail, Dlanpar DzhwarFathullah, Saya Ameer Arsalan Hadi, Shaymaa Faruq Hamadameen, "Melanoma classification using enhanced fuzzy clustering and DCNN on dermoscopy images". NeuroQuantology, 12, 2022, Pages 196-213.
26. Hamadamen, NI, Abdulhameed, SJ, Abdulkrim, HK. Production survey of designing a hybrid optical amplifier (HOP) for WDM systems by using EDFA and Raman amplifiers. *J Des Eng* 2021;7:7476–91.
27. Hamadamen, N. (2021, December 31). Performance Evaluation of WDM Optical Fiber Communication System in the presence of PMD. UKH Journal of Science and Engineering, 5(2), 90-103. https://doi.org/https://doi.org/10.25079/ukhjse.v5n2y2021.pp90-103.
28. Balambica, V. (2021). Static Stress Analysis of an Addendum Modified Spur Gear Pair using FRP Material. *Design Engineering*, 3562-3573.

29. Dr. V. Balambica, Nawroz I. Hamadamen, Dr. A. Karthikayen, M. Praveen, Mr.L. Ganesh Babu, Dr. M. Achudhan, Mr.Dhruv Sangal, Mr.Vishwa Deepak,. "Digital signal processing dual tone multifrequency detector." YMER ,ISSN : 0044-0477, Volume 22 : Issue 02 (2023): 1119-1145.
30. Qaysar Salih Mahdi, Idris Hadi Saleh, Ghani Hashim, Ganesh Babu Loganathan, "Evaluation of Robot Professor Technology in Teaching and Business", Information Technology in Industry, Volume 09, Issue 01, PP 1182 -1194.
31. Dr. Qaysar Salih Mahdi , Dr. Ismail Musa Murad , Ganesh Babu Loganathan. (2022). Prediction Of 3D Digital Map Coverage For UHF Wireless Radio Performance Under Multipath Propagation. *Journal of Pharmaceutical Negative Results*, 9041–9051. https://doi.org/10.47750/pnr.2022.13.S09.1057.
32. Ganesh Babu Loganathan, Amani Tahsin Yasin, "Identification of chromatographical characteristics of complicated biological feeds," Materials Today: Proceedings, Volume 66, Part 3,2022, Pages 1247-1254, ISSN 2214-7853,https://doi.org/10.1016/j.matpr.2022.05.118.
33. Ganesh Babu Loganathan, Qaysar S. Mahdi, Idris Hadi Saleh. (2023). Development Of 5g And Beyond Technology: Challenges & Innovations. *Journal of Pharmaceutical Negative Results*, 1312–1324. https://doi.org/10.47750/pnr.2023.14.S02.159.
34. A.D. Dhass, Ganesh Babu L., Raghuram Pradhan, G.V.K Murthy, M. Sreenivasan ;Energy Harvesting Through Thermoelectric Generators, Materials and Technologies for a Green Environment (2023) 1: 32. https://doi.org/10.2174/978981505121612301000 4.
35. Amarendranath Choudhury, Sathish E, Dhilleshwara Rao Vana, L. Ganesh Babu. "IoT-Based Wrist Attitude Sensor Data for Parkinson's Disease Assessment for Healthcare System." *Practical Artificial Intelligence for Internet of Medical Things*. Ed. Chinmay Chakraborty, Faris A. Almalki Ben Othman Soufiene. Abingdon: CRC PRESS-TAYLOR & FRANCIS GROUP, 2023. 151-168.
36. Dr.Subasini Uthirapathy, M. L. (2023). Screening Of Central Analgesic Activity Of Calotropis Gigantea Flower Using Rats,12 (S3). *European Chemical Bulletin (ISSN 2063-5346)*, 1384 – 1395. DOI: 10.31838/ecb/2023.12.s3.154**.**

37. Selvam, R., Babu, L. G., Thomas, J., Prakash, R., Karthikeyan, T. et al. (2023). Analysis of a Cashew Shell and Fly Ash Rich Brake Liner Composite Material. FDMP-Fluid Dynamics & Materials Processing, 19(3), 569–577.

38. L. Karthick, V. Senthil Murugan, Stephen Leon Joseph Leon, Mahesh Mallampati, M. Ijas Ahamed, Ganesh Babu Loganathan,"Energy performance of a compression refrigeration cycle using environment-friendly refrigerants", Materials Today: Proceedings, Volume 66, Part 3, 2022, Pages 1519-1525, ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2022.07.178.

39. Loganathan, G., Sivam Sundarlingam Paramasivam, S., Kumaran, D., Saravanan, K. et al., "Experimental Study on Verification of Alloy ASTM A510 High-Speed Micro Turning by Parameters Validation through Ranking Algorithm," SAE Technical Paper 2019-28-0071, 2019, https://doi.org/10.4271/2019-28-0071.

40.Mr. Ganesh Babu Loganathan, Dr.Idris Hadi Salih, Dr. Ismail Musa Murad, Dr. Qaysar S. Mahdi, Mr. Qusay Hamed Ali. "Secure cloud storage using blockchain for decentralized system with merkle tree algorithm ." *YMER* 22.4 (2023): 870-893. https://ymerdigital.com/uploads/YMER2204E8.pdf.