Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 9, Issue. 7, July 2020, pg.43 – 53

An Innovate Approach on Biometric and Speech Recognition using Recurrent Neural Networks (RNN)

Shaik Mohammad Ayesha M.Tech(Ph.D)
Assistant Professor (c), IIIT Nuzvid
Department of Computer Science Engineering
Rajiv Gandhi University of Knowledge Technologies
AP-IIIT Nuzvid, Krishna(Dist)
Email: shaikmd.ayesha@gmail.com, Phone: 6304723450

ABSTRACT: *The main objective of this research paper is the use of recurrent neural networks (RNNs) to arise a productive solution to problems in authentication with electrocardiogram (ECG)-based biometrics and Speech Recognition.This paper investigates deep recurrent neural networks, which combine the multiple levels of representation that have proved so effective in deep networks with the flexible use of long range context that empowers RNNs. When trained end-to-end with suitable regularisation.A recurrent neural network (RNN) is a class of Artificial Neural Networks where connections between nodes form a directed Graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected Handwriting and Speech Recognition. Recurrent neural networks (RNNs) are a powerful model for sequential data. Recurrent Neural Networks were designed to work with sequence prediction problems. Sequence prediction problems come in many forms and are best described by the types of inputs and outputs supported.Different RNN architectures with various parameter settings were evaluated, including traditional, long short-term memory (LSTM), gated recurrent unit (GRU), unidirectional, and bidirectional networks. Unlike many existing methods, the RNN-based method does not require any feature extraction.*

Index Terms— Recurrent neural networks, deep neural networks, speech recognition, electrocardiogram (ECG)-based biometrics

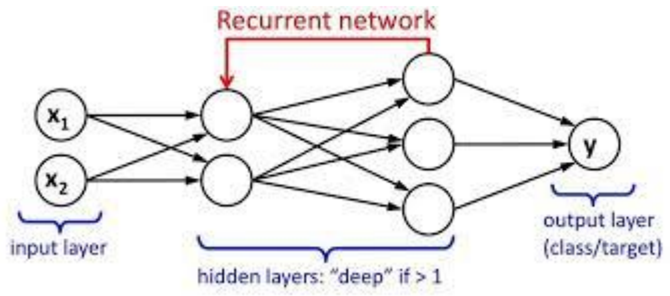

Fig. 1. Long Short-term Memory Cell

INTRODUCTION

Neural networks have a long history in speech recognition, They have gained attention in recent years with the dramatic improvements in acoustic modelling yielded by deep feedforward networks[1]. Given that speech is an inherently dynamic process, it seems natural to consider recurrent neural networks (RNNs) as an alternative model. HMM-RNN systems have also seen a recent revival, but do not currently perform as well as deep networks. Instead of combining RNNs with HMMs, it is possible to train RNNs 'end-to-end' for speech recognition. This approach exploits the larger state-space and richer dynamics of RNNs compared to HMMs, and avoids the problem of using potentially incorrect alignments as training targets. The combination of Long Short-term Memory, an RNN architecture with an improved memory, with end-to-end training has proved especially effective for cursive handwriting recognition. However it has so far made little impact on speech recognition.[2]

Electrocardiogram (ECG)-based biometric systems have been used for authentication. For authentication: there is a declared identity associated with the input data and the system must either accept or reject the declared identity. In this paper, we propose the use of recurrent neural networks (RNNs) to develop an effective solution to problems in ECG-based biometrics. Unlike many methods, the RNN-based method does not require any feature extraction. The ECG data is directly fed to the RNN.

1. SPEECH RECOGNITION WITH DEEP RECURRENT NEURAL NETWORKS

1.1: RECURRENT NEURAL NETWORKS

Given an information grouping x = (x1, . . . , xT), a standard intermittent neural system (RNN) registers the concealed vector arrangement h = (h1, . . . , hT) and yield vector succession y = (y1, . . . , yT) by emphasizing the accompanying conditions from t = 1 to T:

where the W expressions indicate weight grids (for example Wxh is the information shrouded weight network), the b terms mean predisposition vectors (for example bh is shrouded inclination vector) and H is the concealed layer work.

$$h_t = \mathcal{H}\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right) \qquad (1)$$
$$y_t = W_{hy}h_t + b_y \qquad (2)$$

H is normally an elementwise use of a sigmoid capacity. Anyway we have discovered that the Long Short-Term Memory (LSTM) engineering, which uses reason constructed memory cells to store data, is better at finding and abusing long range setting. Fig. 1 shows a solitary LSTM memory cell. For the rendition of LSTM utilized in this paper H is executed by the accompanying composite capacity:

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right) \tag{3}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \tag{4}$$

$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \tag{5}$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right) \tag{6}$$

$$h_t = o_t \tanh(c_t) \tag{7}$$

where σ is the logistic sigmoid function, and i, f , o and c are respectively the input gate, forget gate, output gate and cell activation vectors, all of which are the same size as the hidden vector h. The weight matrices from the cell to gate vectors (e.g. W si) are diagonal, so element m in each gate vector only receives input from element m of the cell vector.

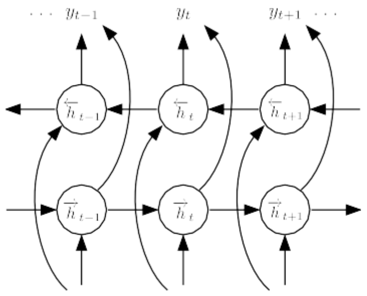
$\cdots\ y_{t-1}$
y_t
$y_{t+1}\ \cdots$
$\overleftarrow{h}_{t-1}$
$\overleftarrow{h}_t$
$\overleftarrow{h}_{t+1}$
$\overrightarrow{h}_{t-1}$
$\overrightarrow{h}_t$
$\overrightarrow{h}_{t+1}$

Fig. 2. Bidirectional RNN

One shortcoming of conventional RNNs is that they are only able to make use of previous context. In speech recognition, where whole utterances are transcribed at once, there is no reason not to exploit future context as well. Bidirectional RNNs (BRNNs) do this by processing the data in both directions with two separate hidden layers, which are then fed forwards to the same output layer. As illustrated in Fig. 2, a BRNN computes the forward hidden sequence h ,the backward hidden sequence h and the output sequence y by iterating the backward layer from t = T to 1, the forward layer from t = 1 to T and then updating the output layer.

$$\overrightarrow{h}_t = \mathcal{H}\left(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right) \quad (8)$$

$$\overleftarrow{h}_t = \mathcal{H}\left(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right) \quad (9)$$

$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y \quad (10)$$

Combing BRNNs with LSTM gives bidirectional LSTM ,which can access long-range context in both input directions. A crucial element of the recent success of hybrid HMM- neural network systems is the use of deep architectures, which are able to build up progressively higher level representations of acoustic data. Deep RNNs can be created by stacking multiple RNN hidden layers on top of each other, with the output sequence of one layer forming the input sequence for the next. Assuming the same hidden layer function is used for all N layers in the stack, the hidden vector sequences h n are iteratively computed from n = 1 to N and t = 1 to T :

$$h_t^n = \mathcal{H}\left(W_{h^{n-1}h^n} h_t^{n-1} + W_{h^n h^n} h_{t-1}^n + b_h^n\right) \quad (11)$$

where we define $\boldsymbol{h}^0 = \boldsymbol{x}$. The network outputs y_t are

$$y_t = W_{h^N y} h_t^N + b_y \quad (12)$$

Deep bidirectional RNNs can be implemented by replacing each hidden sequence h n with the forward and backward sequences h ^{n} and h ^{n} , and ensuring that every hidden layer receives input from both the forward and backward layers at the level below. If LSTM is used for the hidden layers we get deep bidirectional LSTM, the main architecture used in this paper. As far as we are aware this is the first time deep LSTM has been applied to speech recognition, and we find that it yields a dramatic improvement over single-layer LSTM.

1.2: NETWORK TRAINING

We center around start to finish preparing, where RNNs figure out how to outline from acoustic to phonetic successions. One bit of leeway of this methodology is that it evacuates the requirement for a predefined (and blunder inclined) arrangement to make the preparation targets. The initial step is to utilize the system yields to parameterise a differentiable appropriation Pr(y|x) over all conceivable phonetic yield groupings y given an acoustic info arrangement x. The log-likelihood log Pr(z|x) of the objective yield succession z would then be able to be separated regarding the system loads utilizing backpropagation through time , and the entire framework can be enhanced with inclination plunge. We presently depict two different ways to characterize the yield conveyance and consequently train the system. We allude all through to the length of x as T, the length of z as U, and the quantity of potential phonemes as K.

1.3: Connectionist Temporal Classification

The primary technique, known as Connectionist Temporal Classification (CTC) , utilizes a softmax layer to characterize a different yield conveyance Pr(k|t) at each progression t along the info arrangement. This appropriation covers the K phonemes in addition to an additional clear image ∅ which speaks to a non-yield (the softmax layer is along these lines size K + 1). Instinctively the system concludes whether to discharge any mark, or no name, at each timestep. Taken together these choices characterize a dispersion over arrangements between the info and target successions. CTC at that point utilizes a forward-in reverse calculation to aggregate over every conceivable arrangement and decide the standardized likelihood Pr(z|x) of the objective grouping given the info succession. Similar procedures have been used elsewhere in speech and handwriting recognition to integrate out over possible segmentations; however CTC differs in that it ignores segmentation altogether and sums over single-timestep label decisions instead. RNNs trained with CTC are generally bidirectional, to ensure that every Pr(k|t) depends on the entire input sequence, and not just the inputs up to t. In this work we focus on deep bidirectional networks, with Pr(k|t) defined as follows:

$$y_t = W_{\overrightarrow{h}^N y} \overrightarrow{h}_t^N + W_{\overleftarrow{h}^N y} \overleftarrow{h}_t^N + b_y \qquad (13)$$

$$\Pr(k|t) = \frac{\exp(y_t[k])}{\sum_{k'=1}^{K} \exp(y_t[k'])}, \qquad (14)$$

where y t [k] is the k th element of the length K + 1 unnormalised output vector y t , and N is the number of bidirectional levels.

1.4: RNN Transducer

CTC describes an assignment over phoneme groupings that depends just upon the acoustic data progression x. It is consequently an acoustic-simply model. A continuous increment, known as a RNN transducer combines a CTC-like framework with an alternate RNN that predicts each phoneme given the previous ones, thusly yielding a commonly arranged acoustic and language model. Joint LM-acoustic planning has exhibited helpful in the past for talk affirmation.

While CTC chooses a yield flow at each info timestep, a RNN transducer chooses an alternate scattering Pr(k|t, u) for each blend of data timestep t and yield timestep u. In like manner with CTC, each transport covers the K phonemes notwithstanding. Normally the framework 'picks' what to yield depending both on where it is in the data progression and the yields it has quite recently transmitted. For a length U target progression z, the absolute plan of T U decisions together chooses a dissemination over each and every comprehensible course of action among x and z, which would then have the option to be fused out with a forward backward count to choose log Pr(z|x).

In the original formulation Pr(k|t, u) was defined by taking an ‘acoustic’ distribution Pr(k|t) from the CTC network, a ‘linguistic’ distribution Pr(k|u) from the prediction network, then multiplying the two together and renormalising. An improvement introduced in this paper is to instead feed the hidden activations of both networks into a separate feed forward output network, whose outputs are then normalised with a softmax function to yield Pr(k|t, u). This allows a richer set of possibilities for combining linguistic and acoustic information, and appears to lead to better generalisation. In particular we have found that the number of deletion errors encountered during decoding is reduced. Denote by h N and h N the uppermost forward and backward hidden sequences of the CTC network, and by p the hidden sequence of the prediction network. At each t, u the output network is implemented by feeding h N and h N to a linear layer to generate the vector l t , then feeding l t and p u to a tanh hidden layer to yield h t,u , and finally feeding h t,u to a size K + 1 softmax layer to determine Pr(k|t, u):

$$l_t = W_{\overrightarrow{h}^N l} \overrightarrow{h}_t^N + W_{\overleftarrow{h}^N l} \overleftarrow{h}_t^N + b_l \qquad (15)$$

$$h_{t,u} = \tanh\left(W_{lh} l_{t,u} + W_{pb} p_u + b_h\right) \qquad (16)$$

$$y_{t,u} = W_{hy} h_{t,u} + b_y \qquad (17)$$

$$\Pr(k|t,u) = \frac{\exp(y_{t,u}[k])}{\sum_{k'=1}^{K} \exp(y_{t,u}[k'])}, \qquad (18)$$

where y t,u [k] is the k th element of the length K + 1 unnormalised output vector. For simplicity we constrained all non output layers to be the same size (| h nt | = | h nt | = |p u | =|l t | = |h t,u |); however they could be varied independently.RNN transducers can be trained from random initial

weights. However they appear to work better when initialised with the weights of a pretrained CTC network and a pre trained next-step prediction network (so that only the output network starts from random weights). The output layers (and all associated weights) used by the networks during pretraining are removed during retraining. In this work we pretrain the prediction network on the phonetic transcriptions of the audio training data; however for large-scale applications it would make more sense to pretrain on a separate text corpus.

1.5: Decoding

RNN transducers can be decoded with shaft search to yield a n-best rundown of up-and-comer interpretations. In the past CTC systems have been decoded utilizing either a type of bestfirst unraveling known as prefix search, or by essentially taking the most dynamic yield at each timestep. In this work anyway we misuse a similar pillar search as the transducer, with the alteration that the yield mark probabilities Pr(k|t, u) don't rely upon the past yields (so Pr(k|t, u) = Pr(k|t)). We discover pillar search both quicker and more powerful than prefix look for CTC. Note the n-best rundown from the transducer was initially arranged by the length standardized log-probabilty log Pr(y)/|y|; in the current work we abstain from the standardization (which possibly helps when there are a lot a larger number of erasures than additions) and sort by Pr(y).

1.6: Regularisation

Regularisation is vital for good performance with RNNs, as their flexibility makes them prone to overfitting. Two regularisers were used in this paper: early stopping and weight noise (the addition of Gaussian noise to the network weights during training). Weight noise was added once per training sequence, rather than at every timestep. Weight noise tends to ‘simplify’ neural networks, in the sense of reducing the amount of information required to transmit the parameters, which improves generalisation.

1.7: EXPERIMENTS

Phoneme acknowledgment tests were performed on the TIMIT corpus. The standard 462 speaker set with all SA records evacuated was utilized for preparing, and a different improvement set of 50 speakers was utilized for early halting. Results are accounted for the 24-speaker center test set. The sound information was encoded utilizing a Fourier-change based channel manage an account with 40 coefficients (in addition to vitality) circulated on a mel-scale, along with their first and second fleeting subsidiaries. Each information vector was in this way size 123. The information were standardized with the goal that each component of the information vectors had zero mean and unit change over the preparation set. Every one of the 61 phoneme names were utilized during preparing and interpreting (so K = 61), at that point mapped to 39 classes for scoring. Note that all investigations were run just a single time, so the difference because of arbitrary weight initialisation and weight commotion is obscure.

All networks were trained using stochastic gradient descent, with learning rate 10−4 , momentum 0.9 and random initial weights drawn uniformly from [−0.1, 0.1]. All networks except CTC-3l-500h-tanh and PreTrans-3l-250h were first trained with no noise and then, starting from the point of highest log-probability on the development set, retrained with Gaussian weight noise (σ = 0.075) until the point of lowest phoneme error rate on the development set. PreTrans-3l-250h was initialised with the weights of CTC3l-250h, along with the weights of a phoneme prediction network (which also had a hidden layer of 250 LSTM cells), both of which were trained without noise, retrained with noise, and stopped at the point of highest log-probability. PreTrans-3l250h was trained from this point with noise added. CTC-3l500h-tanh was entirely trained without weight noise because it failed to learn with noise added. Beam search decoding was used for all networks, with a beam width of 100. The advantage of deep networks is immediately obvious, with the error rate for CTC dropping from 23.9% to 18.4% as the number of hidden levels increases from one to five. The four networks CTC-3l-500h-tanh, CTC-1l-622h, CTC3l-421h-uni and CTC-3l-250h all had approximately the same number of weights, but give radically different results. The three main conclusions we can draw from this are (a) LSTM works much better than tanh for this task, (b) bidirectional LSTM has a slight advantage over unidirectional LSTMand (c) depth is more important than layer size (which supports previous findings for deep networks). Although the advantage of the transducer is slight when the weights are randomly initialised, it becomes more substantial when pretraining is used.

Table 1. TIMIT Phoneme Recognition Results. ‘Epochs’ is the number of passes through the training set before conver- gence. ‘PER’ is the phoneme error rate on the core test set.

NETWORK	WEIGHTS	EPOCHS	PER
CTC-3L-500H-TANH	3.7M	107	37.6%
CTC-1L-250H	0.8M	82	23.9%
CTC-1L-622H	3.8M	87	23.0%
CTC-2L-250H	2.3M	55	21.0%
CTC-3L-421H-UNI	3.8M	115	19.6%
CTC-3L-250H	3.8M	124	18.6%
CTC-5L-250H	6.8M	150	18.4%
TRANS-3L-250H	4.3M	112	18.3%
PRETRANS-3L-250H	4.3M	144	17.7%

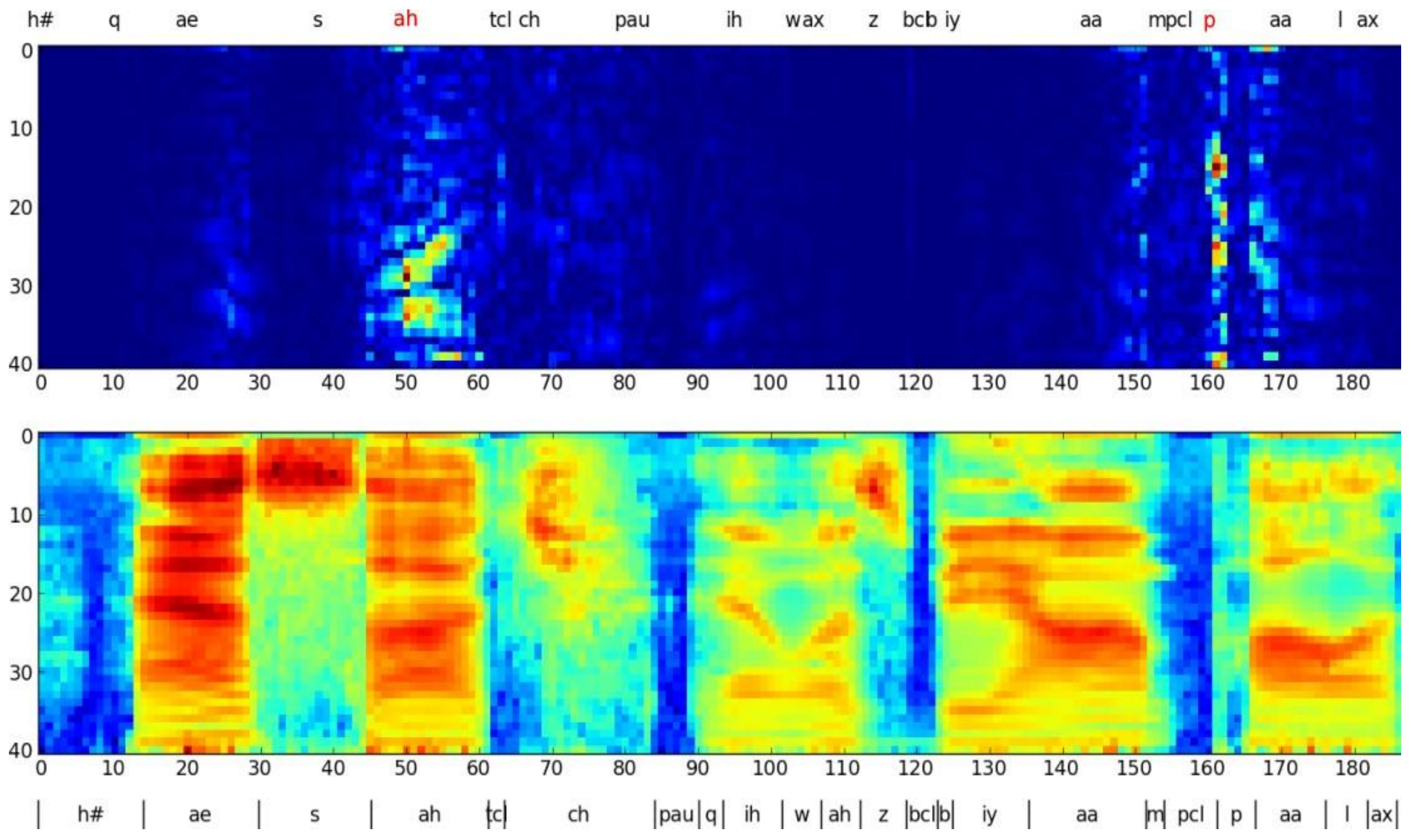

Fig. 3. Input Sensitivity of a deep CTC RNN.

The heatmap (top) shows the derivatives of the 'ah' and 'p' outputs printed in red with respect to the filterbank inputs (bottom). The TIMIT ground truth segmentation is shown below. Note that the sensitivity extends to surrounding segments; this may be because CTC (which lacks an explicit language model) attempts to learn linguistic dependencies from the acoustic data.

LSTM has a slight advantage over unidirectional LSTM and (c) depth is more important than layer size (which supports previous findings for deep networks). Although the advantage of the transducer is slight when the weights are randomly initialised, it becomes more substantial when pretraining is used.

2. ECG-BASED BIOMETRICS USING RECURRENT NEURAL NETWORKS PROPOSED RNN METHOD

2.1: Preprocessing and Segmentation

The ECG accounts utilized in this work are from the openly accessible ECG-ID and MIT-BIH Arrhythmia (MITDB) datasets, which are a piece of the Physionet database. Examination was performed independently for the two datasets. Given an ECG recording, the initial step is to section the chronicle into singular heartbeat waveforms. Since the R top is the most unmistakable pinnacle, it tends to be utilized as a marker of a given heartbeat waveform. The R tops were identified utilizing the Pan-Tompkins calculation. When the pinnacles are distinguished, a specific number of tests when a given R top are connected, shaping a vector which speaks to the heartbeat waveform. For the ECG-ID dataset, 150 examples previously/after the R top were chosen, while 125 examples were chosen for the MITDB dataset. After the division system is finished, every individual heartbeat waveform is z-score normalized. At long last, a specific number of back to back heartbeat waveforms are gathered to shape a given info grouping, where the quantity of pulses in the information arrangement is a hyperparameter.

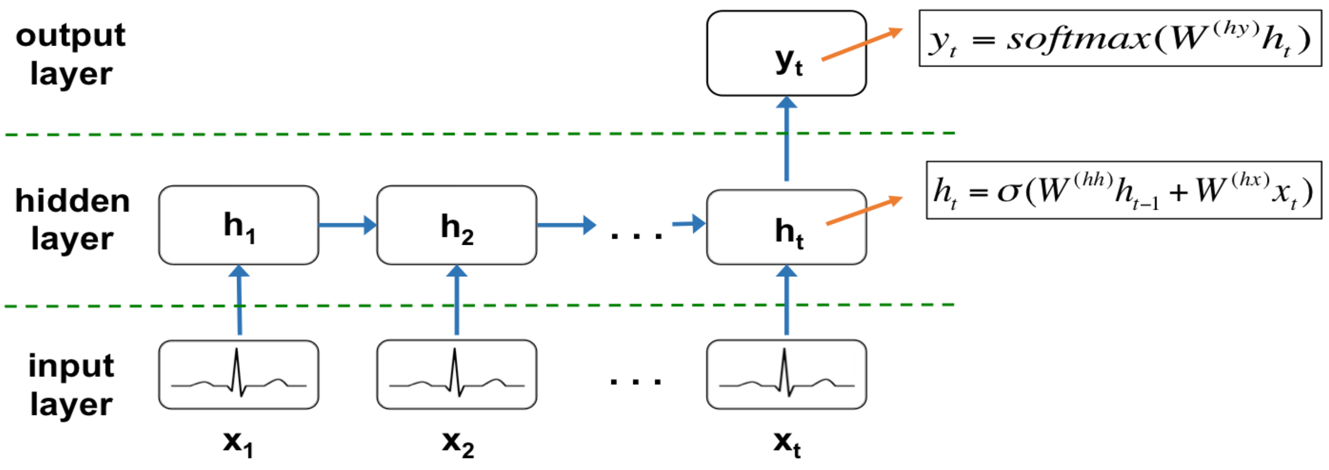
output layer
hidden layer
input layer
y_t
h_1
h_2
h_t
x_1
x_2
x_t
$y_t = softmax(W^{(hy)}h_t)$
$h_t = \sigma(W^{(hh)}h_{t-1} + W^{(hx)}x_t)$

Fig. 4. Block diagram of a traditional RNN applied to an input sequence of *t* heartbeats ($x_1x_2 \ldots x_t$) from a given subject. The hidden state and output at the final time step are denoted as ht and yt, respectively. The weight matrices $W^{(hh)}$, $W^{(hx)}$, and $W^{(hy)}$ are parameters optimized during training, and σ represents an element-wise non-linearity such as *tanh* or ReLU.

2.2 Authentication Procedure

For the verification situation, the ECG accounts are separated into three datasets: preparing, enlistment, and assessment. The subjects utilized during preparing are not quite the same as those utilized during enlistment/assessment. Moreover, the subjects in the enlistment and assessment sets are indistinguishable. Preparing of the RNN is acted in the manner depicted in the past subsection, with the softmax layer present. After the RNN has been effectively prepared, the softmax layer is expelled. The testing stage comprises of two phases: enlistment and assessment. In the enlistment stage, the prepared system is utilized to extricate the rundown vector (for example the concealed state at the last time step) from every enlistment grouping. For each subject, there are numerous enlistment groupings, each creating an alternate outline vector, and the enlistment model of a given subject is taken to be the normal of these vectors. The subjects in the assessment set are indistinguishable from those in the enlistment set, with the exception of new successions are taken care of to the RNN. An assessment succession from a given subject is taken care of to the system, yielding an outline vector.This summary vector is compared to each enrollment model (one for each subject in the enrollment/evaluation set) using a similarity metric (e.g. cosine distance), and an authentication decision for each comparison is made based on a given threshold. The claimed identity is accepted or rejected depending on whether the distance is less than or greater than the given threshold, respectively. The advantage of the summary vector approach is that the network can be trained using a given number of subjects, and testing (enrollment/evaluation) can be performed on any other subjects that were not necessarily part of the training set. In this method, the trained RNN can be viewed as a feature extractor, in which the summary vector represents the extracted feature vector; similar techniques have been used for speaker verification.

2.3: Dataset and Implementation

The ECG-ID dataset contains 310 accounts, got from 90 subjects (44 male and 46 female). Each recording is ECG lead I, recorded over a length of 20 seconds, and digitized at 500 Hz with 12-piece goals over an ostensible ±10 mV run. Two chronicles for each subject were utilized for the examination, in light of the fact that solitary a little subset of the subjects have multiple accounts. The MITDB dataset contains 48 two-channel chronicles, got from 47 subjects (25 male and 22 female). Each subject has just one chronicle accessible, aside from one subject that has two accounts (records 201 and 202); just record 201 was utilized for this subject. The chronicles were digitized at 360 examples for each second per channel with 11-piece goals over a 10mV territory. Just the upper sign was utilized on the grounds that QRS edifices are generally noticeable in the upper sign; a remarkable special case is record 114, for which the signs are turned around. Both the ECG-ID and MITDB datasets contain pre-separated information, which was utilized in this examination. R top identification and age of the preparation and testing information were acted in Matlab, while execution and preparing of RNNs were performed utilizing TensorFlow. The cost work utilized during preparing is the cross-entropy mistake, and improvement was performed utilizing the Adam calculation with a learning pace of 0.0001. We assessed distinctive RNN models with various boundary settings.

Table 2. ECG-ID Within-Session Analysis (Identification): Classification accuracy for selected parameter settings.

Type of cell/unit	Input sequence length (in number of beats)	Number of hidden layers	Classification Accuracy
Traditional	3	1	92.8%
Traditional	9	1	93.3%
GRU	3	1	95.2%
GRU	9	1	96.7%
LSTM	3	1	98.2%
LSTM	3	2	97.8%
LSTM	9	1	100%

The architectures we tested are LSTM, GRU, and traditional RNNs, and we tested both unidirectional and bidirectional architectures. The RNN parameters are the number of layers, number of cells or units per layer, and dropout rate. Hyperparameter optimization was performed using random search. For the identification problem, the classification accuracy was computed using the testing set. A classification decision is made for each testing sequence, and the classification accuracy is defined to be the percentage of correctly classified testing sequences. For the authentication scenario, the equal error rate (EER) was computed in the following manner. Given a summary vector obtained from the evaluation data of a particular subject, this vector is compared to each enrollment model (one model for each subject in the enrollment/evaluation set) using cosine distance as a similarity metric. An authentication decision for each comparison is then made based on a given threshold. Thus, for each summary vector obtained from the evaluation dataset, there is a certain number of each of the following quantities: true acceptance, true rejection, false acceptance, false rejection. Each of these 4 quantities is summed across the set of summary vectors obtained from the evaluation dataset, and the false acceptance rate (FAR) and false rejection rate (FRR) are computed. The threshold is then varied to yield a plot of FAR and a plot of FRR, and the EER is taken to be the intersection of these two curves.

2.4: Authentication Analysis

For the ECG-ID dataset, 9 beats were taken from every meeting for a given subject in the preparation set, for an aggregate of 18 preparing beats for each subject, while for each subject in the enlistment/assessment set, 18 enlistment beats were taken from one meeting and 18 assessment beats were taken from the other meeting. For the MITDB dataset, 18 beats were taken for a given subject in the preparation set, and for each subject in the enlistment/assessment set, 18 beats were taken for enlistment and 18 beats were taken for assessment. Likewise, we assessed the impact of shifting the level of subjects utilized for preparing. The outcomes demonstrated are for input successions comprising of 9 pulses. The figure shows that the EER diminishes as the preparation size increments, and besides, that the proposed strategy can accomplish 0% EER when the level of subjects utilized for preparing is roughly 80%.

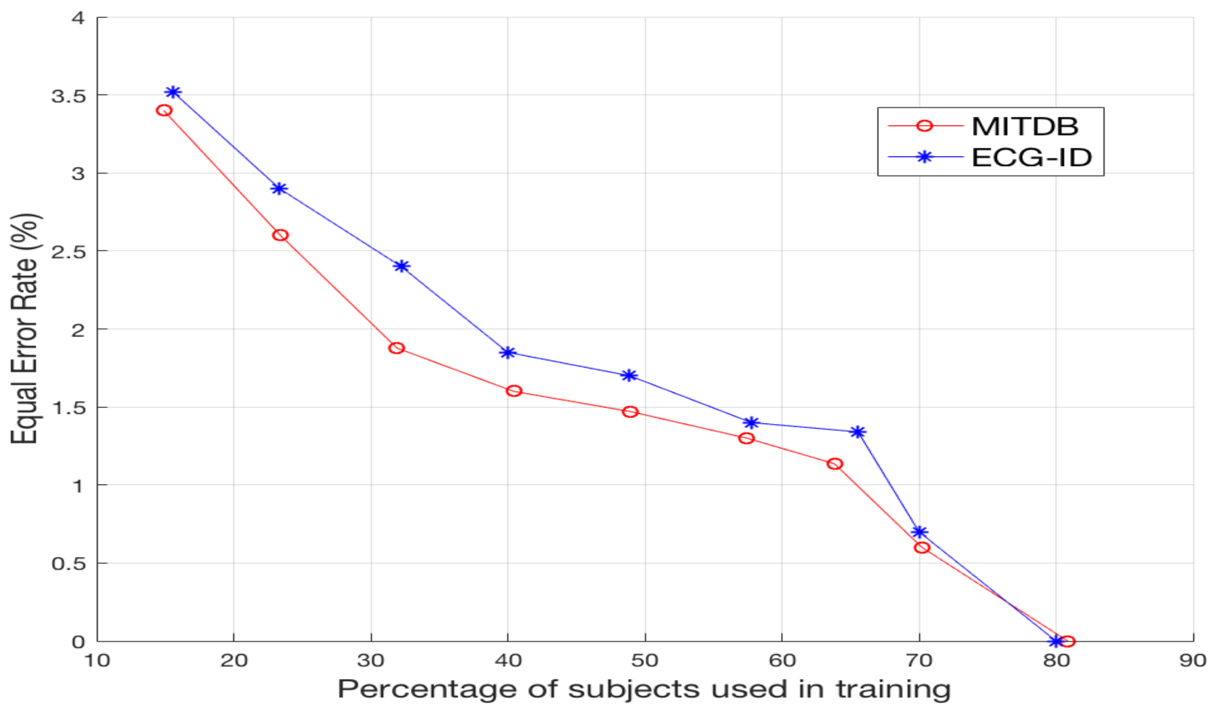

Fig. 5. **Authentication analysis: Equal error rate (%) as a function of the percentage of subjects used in training, for the ECG-ID and MITDB datasets.**

CONCLUSION

We have demonstrated that (1) the combination of deep, bidirectional Long Short-term Memory RNNs with end-to-end training and weight noise gives state-of-the-art results in phoneme recognition on the TIMIT database and (2) that an LSTM-based RNN is a more productive tool for ECG-based biometric authentication, compared to methods that uses the ECG-ID or MITDB datasets. We estimate the effect of the training size on the EER, and found that the EER drops to 0% when the percentage of subjects used for training is increased to approximately 80%. Our work proves that LSTM-based RNNs are encouraging direction for ECG-based biometric authentication.

ACKNOWLEDGEMENT

Author is thankful to the director of Apiit, Nuzvid, ap, india, for providing research facilities for the entire work.

REFERENCES

[1] Lee, Li and Rose, R. A frequency warping approach to speaker normalization. Speech and Audio Processing, IEEE Transactions on, 6(1):49–60, Jan 1998.

[2] Qifeng Zhu, Barry Chen, Nelson Morgan, and Andreas Stolcke, “Tandem connectionist feature extraction for conversational speech recognition,” in International Conference on Machine Learning for Multimodal Interaction, Berlin, Heidelberg, 2005, MLMI’04, pp. 223– 231, Springer-Verlag.