Recurrent neural network-based speech recognition using MATLAB

Praveen Edward James, Mun Hou Kit* and Chockalingam Aravind Vaithilingam

School of Engineering,
Taylor's University,
Taylor's University Lakeside Campus,
No. 1, Jalan Taylor's, 47500 Subang Jaya,
Selangor, Malaysia
Email: PraveenEdwardJames@sd.taylors.edu.my
Email: HouKit.Mun@taylors.edu.my
Email: ChockalingamAravind.Vaithilingam@taylors.edu.my
*Corresponding author

Alan Tan Wee Chiat

Multimedia University,
Jalan Ayer Keroh Lama, 75450 Bukit Beruang,
Melaka, Malaysia
Email: wctan@mmu.edu.my

Abstract: The purpose of this paper is to design an efficient recurrent neural network (RNN)-based speech recognition system using software with long short-term memory (LSTM). The design process involves speech acquisition, pre-processing, feature extraction, training and pattern recognition tasks for a spoken sentence recognition system using LSTM-RNN. There are five layers namely, an input layer, a fully connected layer, a hidden LSTM layer, SoftMax layer and a sequential output layer. A vocabulary of 80 words which constitute 20 sentences is used. The depth of the layer is chosen as 20, 42 and 60 and the accuracy of each system is determined. The results reveal that the maximum accuracy of 89% is achieved when the depth of the hidden layer is 42. Since the depth of the hidden layer is fixed for a task, increased performance can be achieved by increasing the number of hidden layers.

Keywords: speech recognition; feature extraction; pre-processing; recurrent neural network; RNN; long short-term memory; LSTM; hidden layer; MATLAB.

Reference to this paper should be made as follows: James, P.E., Kit, M.H., Vaithilingam, C.A. and Chiat, A.T.W. (2020) 'Recurrent neural network-based speech recognition using MATLAB', *Int. J. Intelligent Enterprise*, Vol. 7, Nos. 1/2/3, pp.56–66.

Biographical notes: Praveen Edward James completed his BEng from Madurai Kamaraj University, Madurai, India and MTech from SRM University, Chennai, India in 2002 and 2005 respectively. He worked as a project consultant in a sole proprietorship concern in Tuticorin, India, from 2006 to 2012. He also worked as a Lecturer and Coordinator for the School of Computing at Olympia college, Kuala Lumpur, Malaysia from 2012 to 2014.

His research interests include data science, FPGA-based prototyping, internet of things (IoT), hidden Markov models, deep learning, speech processing and robotics. He is currently pursuing his PhD in Engineering from Taylor's University, Subang Jaya, Malaysia.

Mun Hou Kit received his BEng (Hons.), MEng, and PhD degrees in Electrical Engineering from the Universiti Teknologi Malaysia, Skudai, Johor, Malaysia, in 2006, 2008, and 2014, respectively. He is currently a Lecturer with the Electrical and Electronic Engineering Program, School of Engineering, Taylor's University, Subang Jaya, Selangor, Malaysia. His current research interests include microwave sensor and applications, material dielectric properties measurement, field-programmable gate array (FPGA) and the internet of things (IoT) implementation as well as machine learning, electrocardiogram (ECG) and speech processing. He is a registered member and graduate engineer with the Institution of Engineering and Technology (IET) and Board of Engineers Malaysia (BEM), respectively. He is also registered with the Engineering Council of the United Kingdom, as a Chartered Engineer.

Chockalingam Aravind Vaithilingam received his BEngg degree from Bharathidasan university in 1998, MEngg from Bharthiyar University in 2001 and PhD degree in Electrical Power Engineering from University Putra Malaysia in 2013. From 2003 till date he worked as key researcher in the design of energy efficient and special electrical machines. Since 2011, he has been a Senior Lecturer with the Electrical and Electronics Engineering Department, Taylor's University, Malaysia. He is the author of five books, more than 150 articles, and four inventions. His research interests include novel energy efficient machines for transportation and alternative energy applications. He holds four filed application patents and one approved patent.

Alan Tan Wee Chiat completed his BEng (Hons), MEng and PhD from Multimedia University. He is currently providing services as an Associate Professor at Multimedia University, Melaka, Malaysia. He has authored and co-authored multiple peer-reviewed scientific papers and presented works at many national and International conferences. He contributions have acclaimed recognition from honourable subject experts around the world. He is actively associated with different societies and academies. His academic career is decorated with several reputed awards and funding. His research interests include digital signal processing, digital image processing, pattern classification and baseband communications.

1 Introduction

Speech processing has continuously evolved over the years. State of the art systems is continuously replaced over time. In general, speech processing involves the following: a recogniser or a speech-to-text module that converts speech signals into text, a parser that extracts the semantic context, a dialog manager that determines system response in machine language, an answer generator that provides the system response in text and a speech synthesiser that converts text to the speech signal.

Speech processing units may be stand-alone or combined to form different applications. The outcomes of these processes are designed such as Siri, Cortana, Google

Voice Search and so on. A collection of such different speech processing applications is given in Table 1.

Table 1 Existing speech applications

No.	*Application*	*Manufacturer*
1	Voice Search	Google Inc.
2	Vlingo Virtual Assistant	Vlingo Corporation
3	Iris (alpha)	Dexetra
4.	Speaktoit Assistant	Speaktoit
5	Skyvi	Blue Tornado
6	AIVC	YourApp24
7	Car Home	Google Inc.
8	Dragon Search	Nuance Communications
9	Voice Actions/ Jeanie	Pannous
10	Everfriends	i-Free Innovations
11	Evi	True Knowledge Ltd
12	Andy-Siri for Android	74Technologies
13.	Edwin, Speech-to-Speech	Neureau
14	Dragon Go	Nuance Communications
15	Speak 4it	AT&T Interactive R&D
16	Voice-Assistant	Quantic Apps
17	Pocket Blonde	i-Free Innovations
18	EVA Virtual Assistant	BulletProof
19	Ziplocal	Phone Directories Company
20	Chizee Your Personal Assistant	Tronton LLC

Source: Husnjak et al. (2014)

Speech recognition systems usually involve classification of acoustic templates with pre-known classes. Some of the algorithms for speech recognition includes dynamic time warping (DTW) (Mohan, 2014), hidden Markov model (HMM) (Sha and Saul, 2006) Gaussian mixture model (GMM) (Vyas, 2013), subspace Gaussian mixture models (SGMMs) (Ghalehjegh and Rose, 2013) support vector machines (SVMs) (Ganapathi, 2002) and deep neural networks (DNNs) (Graves et al., 2013).

Long short-term memory (LSTM) requires an optimal number of recurrent units to minimise training time is the current state-of-the-art algorithm. The problem of overfitting results in lower accuracy; hence, this work involves the design of variants of vanilla LSTM-based speech recognition to obtain the optimal model. Various RNN implementations are available in the literature. In Graves et al. (2013) individual RNN models are designed for each RNN computation unit and a discriminative training procedure is used. Each RNN speech model is adjusted to reduce its distance from the designated speech unit. The model is simple in design.

In Venkateswarlu et al. (2011) RNN is used to understand the difference between similar phonemes. The developed model is also compared with a multi-layer perceptron to evaluate their performance metrics. Characters from the phoneme E to AH set are used

as a training and test data. This technique illustrates the discriminative capability of RNN in two levels of processing namely the phonetic and word levels. In Huang et al. (2014) several caches are utilised to minimise computational cost of RNN language model which is more suited for a hardware implementation.

In the paper Graves et al. (2013) a deep recurrent neural network is designed as a multi-level representation for flexible use of long-range context. Graves and Jaitly (2014) proposed LSTM architecture with connectionist temporal classification to achieve direct optimisation of word error rate. The model directly processes speech data from spectrograms and starts from the character level. Wu et al. (2017) propose a novel approach for monitoring and fault-protection of super magnets using LSTM.

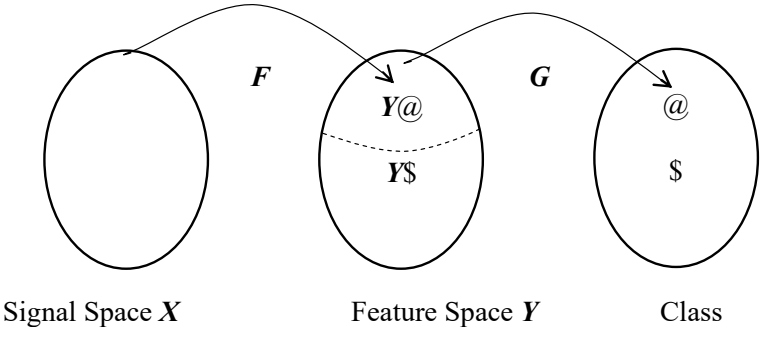
In Chang et al. (2015) vanilla LSTM neural networks are utilised to predict the remaining useful life (RUL) of critical healthcare equipment to prevent critical failure hazards. The model provides optimum performance by utilising LSTMs, in the cases of complicated operations, working conditions, model degradations, and strong noises.

HORNN, a variant of RNN (Zhang and Woodland, 2018) reduces the complexity of LSTM but uses more connections from previous time steps to eliminate vanishing long-term gradients with additional processing time. While many implementation styles are available, vanilla LSTM is chosen because of the design simplicity, optimum storage, training time and its ability to eliminate long-range dependency problem. This paper is divided into the following sections: Section 2 reveals background information behind the structure and working of LSTM. Section 3 presents the adopted design methodology. Section 4 provides the results and its analysis. Section 5 concludes the topic with recommendations.

2 Background

Pattern recognition involves identifying the best match for the unknown signal from known signal classes. LSTM-based speech recognition predicts an unknown signal based on the probability of occurrence of pre-determined model parameters. It is a signal classification problem which is illustrated in Figure 1.

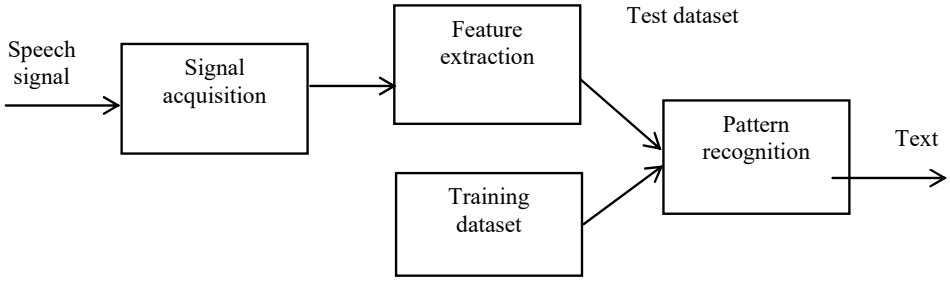
Figure 1 Pattern recognition



Here Y is partitioned into many non-overlapping regions. G involves mapping the partitioned regions (Y_c) to class labels (G: Y_c -> C). This process requires increased processing field as the number of class labels increases and can be efficiently

implemented using RNNs. The choice of the implementation style can be proposed after conducting experiments on using RNNs for speech recognition. In general, speech recognition involves the process of converting speech signals into text. The speech signals may be represented as feature vectors of a word or phoneme. The block diagram of the overall system is obtained by incorporating the tasks into various processing stages and is given in Figure 2.

Figure 2 Block diagram of a speech recognition system



2.1 Signal acquisition

Speech signals are captured into the system using an analogue to digital converter (ADC) to digitise the signal. The signal is passed through a band-pass filter to eliminate low and high-frequency noise and a pre-emphasis filter to boost the high-frequency content of the signal. While designing as a hardware-software co-design this process can be performed by using a software with the support of an audio coder-decoder (CODEC).

2.2 Feature extraction

The process involves the generation of features to uniquely represent a speech signal. Mel frequency Cepstral coefficients (MFCC) features are commonly used in speech recognition. Initially, the signals are passed through a filter bank to obtain the sub-bands corresponding to the frequencies. The Mel values are then calculated using equation (1).

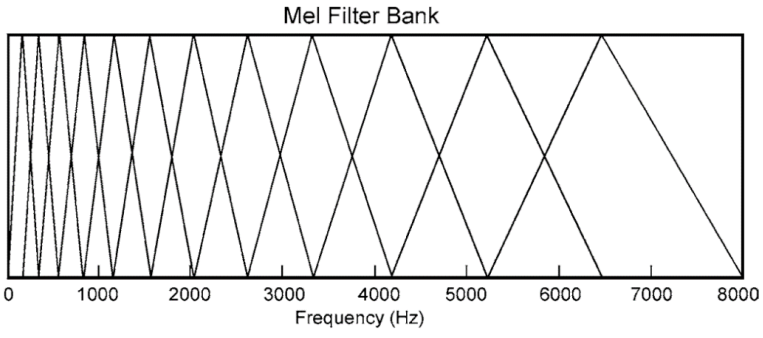
$$mel = 2,595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{1}$$

A Mel to frequency plot is shown in Figure 3. The Cepstral coefficients are calculated by determining the spectrum of spectrum m of a signal. The number of coefficients may be varied to achieve variations in accuracy.

2.3 Training phase

It involves pre-processing and feature extraction of signals with known classes. This data acts as the template for pattern recognition. The training involves repeated calculations and hence not suitable for hardware implementation. This is achieved by training software and stored in any of the memory spaces available. This achieves an increased throughput of the design which is the time taken for repeated outputs.

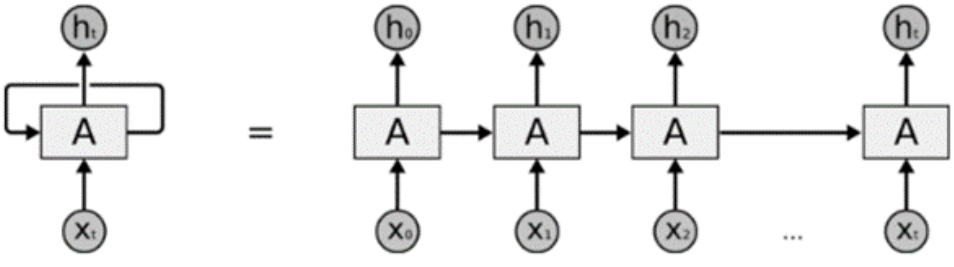
Figure 3 Mel to frequency plot



2.4 RNN structural description

Humans understand each word based on the understanding of previous words. Human thoughts have persistence. Traditional neural networks can not do this but recurrent neural networks (RNN) address this issue. They are networks with loops in them, allowing information to persist. The interpretation of their structure is given in Figure 4 (Colah's Blog, 2015).

Figure 4 RNN structure



A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. It has a memory unit whereby time-series data like speech can be processed sequentially and stored. They can go back in time to recollect the previous data. This makes them extremely useful for speech recognition.

The depth of the RNN is vital to the ability to process sequential data and multiple layers enable them to learn multiple layers of representation. There are two equations which characterise the computation of RNN as given by equation (2) and equation (3).

$$h^{(t)} = \sigma\left(Ux^{(t)} + Wh^{(t-1)}\right) \tag{2}$$

$$Y^{(t)} = softmax\left(Vh^{(t)}\right) \tag{3}$$

where

$x^{(t)}$ is the input unit at time t

$h^{(t)}$ is the hidden unit at time t

$Y^{(t)}$ is the output unit at time t

σ is the activation function called the sigmoid function

softmax is a statistical function to calculate the maximum value

U, V and W are the weights of the input, output and hidden units respectively.

The input signal at time t is provided to the input unit which is multiplied by u and transferred to the hidden unit as current the value. The hidden unit stores previous state $h^{(t-1)}$ and is accumulated with the current value after multiplication with weight W limited by a sigmoid function. The output unit calculates the maximum of the hidden values to obtain an output (Venkateswarlu et al., 2011).

An integral part of RNN is the training algorithm. Training involves updating weights based on inputs with target outputs. It involves computing a cost function or error function between actual output and target output and utilising a mechanism to minimise the cost function. Back propagation with cross-entropy cost function is a widely adopted technique (Tang and Skorin-Kapov, 2001).

The cross-entropy loss function is defined as in equation (4).

$$E(y, \hat{y}) = \sum_t y_t \log \hat{y}_t \tag{4}$$

The gradients of weights are calculated recursively. Considering the weight of the hidden unit the gradient is calculated as in equation (5)

$$\frac{\partial E}{\partial W} = \sum_t \frac{\partial E_t}{\partial W} \tag{5}$$

Let us consider $t = 3$. We obtain equation (6).

$$\frac{\partial E_s}{\partial W} = \frac{\partial E_s}{\partial \hat{Y}_s} \frac{\partial \hat{Y}_s}{\partial h_s} \frac{\partial h_s}{\partial W} \tag{6}$$

3 Methodology and framework

LSTM networks, a category of RNNs handle arbitrarily long sequences to perform speech recognition. An LSTM structure is shown in Figure 5 (Colah's Blog, 2015).

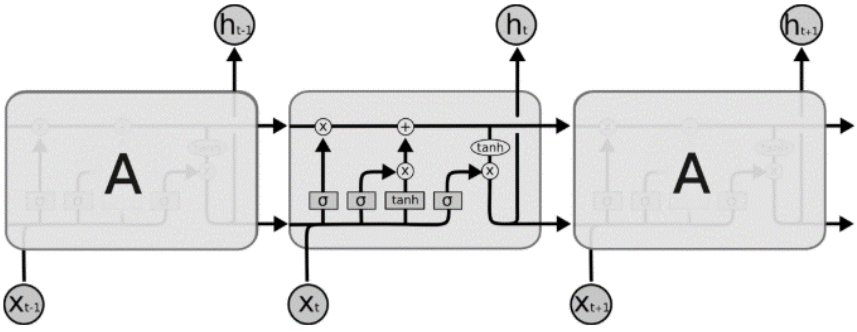
In the structure, a combination of input tex_t and previous hidden state value h_{t-1} is combined and squashed between –1 and 1 by *tanh* function and allowed into the network (i_t) by the input gate implemented by a sigmoid function with values between 1 and 0.

The values are then stored in a memory cell (c_t) controlled by the forget gate implemented by a sigmoid function. When the forget gate is close to 1, it accumulates the current and previous cell value (c_{t-1}) to obtain the hidden layer output (h_t). When the forget gate is close to 0, the previous values are erased. The output from the cell (ot) is squashed again by a *tanh* function and is transferred to the output controlled by a sigmoid-based output gate.

Some of the variants of vanilla LSTM include one-hot encoded LSTM (Lu et al., 2015). LSTM with word embeddings (He et al., 2016) and binary encoded LSTMs. In this implementation, binary encoding is used. In this type, all the active words in a

sentence are given a binary value '1', while all other words are assigned the value '0'. This enables faster processing and convergence.

Figure 5 Structure of an LSTM layer



The entire system is designed using MATLAB. Speech signals are pre-processed and MFCC features are extracted to form the training dataset. Then the LSTM layer is trained on the dataset and is mapped to the word classes. In real time the test signal is captured, pre-processed and features are extracted. These features are classified by the trained network according to the pre-defined classes. These results are calculated in a system with Intel i7 550U 2.40 Processor and 8 GB RAM. The training signals are vectorised using a binary encoding processed and passed to the LSTM network. These values are compared with target classes and the weights are updated during the training phase. The test signals are then passed to the network and their target values are predicted based on the trained weights.

4 Results and analysis

The LSTM network uses a depth of 42 in the hidden layer and there are 20 sentences used with a total of 81 words. During training, acoustic elements of a speech signal are generated and processed to obtain 20 observations which are fed to a single RNN layer. There are 20 output units based on the class labels. Back propagation is used to calculate the weight gradients and the weights are updated by minimising a cost function. The parameters that are to be taken into consideration are the throughput and accuracy. These results are obtained by varying the depth of the hidden layer. The accuracy of each word is calculated from Table 2.

It is calculated by taking the average of the recognised sentence with every sentence spoken ten times. The overall accuracy of each system is also calculated by averaging the accuracy of all sentences. The variation in the accuracy of the system with respect to the depth of the hidden layer is shown in Figure 6.

The processing time and the accuracy of the system are calculated for different depths and the results are shown in Table 3. From the table, it can be found that there is an increase in training time when the depth of the hidden layer increases and the accuracy also increases to a certain extent then decreases. This is due to the problem of overfitting.

The system with a depth of 20 has the least processing time of 13.41 seconds but the accuracy is 83.5%. The system with the depth of 42 has the maximum accuracy at 89%

along with a processing time of 17.13 seconds. The system with a depth of 60 has an accuracy of 86.5% with a processing time of 22.38. The second system is used for final recognition.

Table 2 Accuracy of the speech recognition system

No.	*Sentences used*	*Hidden layer depth-20*	*Hidden layer depth-42*	*Hidden layer depth-60*
1	Samsung founder is	9	9	9
2	Where is Samsung located	9	9	9
3	Greet everyone	7	8	8
4	Where are we	6	7	6
5	Call Dr. Aravind	8	9	7
6	What is your name	7	8	7
7	The main competitor of Samsung	9	10	9
8	Send email to Praveen	8	8	8
9	Turn on the camera	7	8	7
10	Conclude the event	8	9	9
11	what is the OS of Samsung	9	10	10
12	Tell the RAM size of S6	9	9	9
13	S6 was released in	8	9	9
14	The cost of S6 is	9	10	10
15	Do you own S6	7	7	8
16	Flights to Boston today	10	10	10
17	Next flight to Perth	9	9	9
18	Is flight AK021 delayed	8	9	9
19	Send message to Dr. Mun	10	10	10
20	Are seats available	10	10	10
Average WER		8.35	8.9	8.65

Figure 6 Speech recognition system accuracy (see online version for colours)

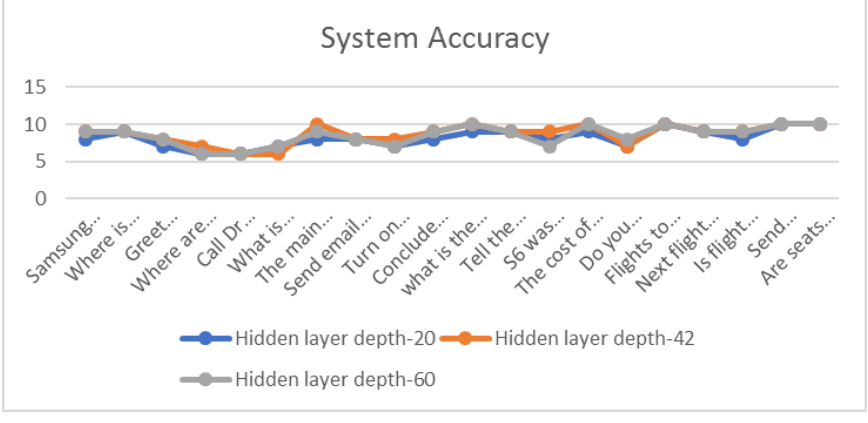


Table 3 Accuracy and processing time of the recognition system with depth variations

Parameters	*Depth (20)*	*Depth (42)*	*Depth (60)*
Processing time	13.41s	17.13s	22.38s
Accuracy (%)	83.5	89	86.5

This is a vanilla type LSTM. Once the number of hidden units is fixed, increased performance can be achieved by increasing the number of hidden layers. However, the processing time increase as the number of layers increase which can be resolved using a dedicated hardware device. For comparison, speech recognition models with their accuracy are given in Table 4.

Table 4 Models comparison -accuracy

No.	*Model used*	*Accuracy (%)*
1	Hidden Markov model	77.35
2	Gaussian mixture model	78.3
3	Deep neural network	79.5
4	Subspace Gaussian mixture model	72.3
5	Support vector machine	73
6	Recurrent neural network	89

Source: Hinton et al. (2012)

5 Conclusions and future work

The software implementation in MATLAB has provided an insight into the speech recognition process. The accuracy of the system is of utmost importance while the processing time also plays a major role. The accuracy of the system can be enhanced by using multiple layers at the cost of additional processing time. The network, on the other hand, has superior performance when realised using dedicated hardware with the parameters pre-determined. Some of the factors to be considered for hardware are a serialisation of input and output data, performing multiplication and division using repeated additions and subtractions, using fixed-point calculations and avoiding loops, large arrays, and matrices.

References

Chang, A.X., Martini, B. and Culurciello, E. (2015) 'Recurrent neural networks hardware implementation on FPGA', arXiv preprint arXiv: 1511.05552.

Ganapathi, R.A. (2002) *Support vector machines for Speech Recognition*, Doctoral dissertation, Mississippi State University.

Ghalehjegh, S. H. and Rose, R.C. (2013) 'Phonetic subspace adaptation for automatic speech recognition', in *ICASSP*, May, pp.7937–7941.

Graves, A. and Jaitly, N. (2014) 'Towards end-to-end speech recognition with recurrent neural networks', in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp.1764–1772.

Graves, A., Mohamed, A.R. and Hinton, G. (2013) 'Speech recognition with deep recurrent neural networks', *IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 26 May, pp.6645–6649.

He, W., Wang, W. and Livescu, K. (2016) "'Multi-view recurrent neural acoustic word embeddings', arXiv preprint arXiv: 1611.04496.

Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A-R., Jaitly, N. and Kingsbury, B. (2012) 'Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups', *Signal Processing Magazine*, IEEE, Vol. 29, No. 6, pp.82–97.

Huang, Z., Zweig, G. and Dumoulin, B. (2014) 'Cache based recurrent neural network language model inference for first pass speech recognition', *IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 4 May, pp.6354–6358.

Husnjak, S., Perakovic, D. and Jovovic, I. (2014) 'Possibilities of using speech recognition systems of smart terminal devices in traffic environment', *Procedia Engineering*, Vol. 69, pp.778–787.

Lu, L., Zhang, X., Cho, K. and Renals, S. (2015) 'A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition', in *16th Annual Conference of the International Speech Communication Association*.

Mohan, B.J. (2014) 'Speech recognition using MFCC and DTW', *2014 International Conference on IEEE in Advances in Electrical Engineering (ICAEE)*, January, pp.1–4.

Sha, F. and Saul, L.K. (2006) 'Large margin hidden Markov models for automatic speech recognition', in Schölkopf, B., Platt, J.C. and Hoffman, T. (Eds.): *Advances in Neural Information Processing Systems 19: Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 4–7 December, pp..–249–1256, MIT Press, Cambridge.

Tang, K.W. and Skorin-Kapov, J. (2001) 'Training artificial neural networks: backpropagation via nonlinear optimization', *CIT. Journal of Computing and Information Technology*, 30 March Vol. 9, No. 1, pp.1–4.

Colah's Blog (2015) *Understanding LSTM* [online] http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed 4 January 2018).

Venkateswarlu, R.L.K., Kumari, R.V. and JayaSri, G.V. (2011) 'Speech recognition by using recurrent neural networks', *International Journal of Scientific and Engineering Research*, June, Vol. 2, No. 6, pp.1–7.

Vyas, M. (2013) 'A Gaussian mixture model based speech recognition system using MATLAB', *Signal and Image Processing, an International Journal (SIPIJ)*, Vol. 4, No. 4, pp.109–118.

Wu, Y., Yuan, M., Dong, S., Lin, L. and Liu, Y. (2017) 'Remaining useful life estimation of engineered systems using vanilla LSTM neural networks', *Neurocomputing*, Vol. 275, pp.167–179, doi:10.1016/j.neucom.2017.05.063.

Zhang, C. and Woodland, P. (2018) 'High order recurrent neural networks for acoustic modelling'. arXiv preprint arXiv: 1802.08314.