



Speech Recognition Based On Machine Learning

A C Namruthah¹, Amutha V², Jyothi M³, Tamil Varshini E S⁴,

Thaniska L S⁵, Karthika K⁶

^{1, 2, 3, 3, 4, 5, 6} Computer Science and Engineering, Sri Sairam College of Engineering, Anekal, Bengaluru-562106

¹sce23cs006@sairamtap.edu.in, ²sce23cs036@sairamtap.edu.in.

³sce23cs034@sairamtap.edu.in, ⁴sce23cs026@sairamtap.edu.in,

⁵sce23cs031@sairamtap.edu.in, ⁶sugunaa.cse@sairamce.edu.in

ABSTRACT

Speech is the fundamental form of communication between the speaker and the listener. Machines are unable to understand the speaker's words, while humans can easily understand their meaning. Today, machines take up the majority of our time, but we are unable to communicate with them. Speech recognition requires both the human brain and machine learning technology to interface with humans and robots. This paper uses English as the speech recognition language because it is a universal language.

1 Introduction

A layer-based machine learning model called a residual neural network is utilized to construct the pyramidal cells. Pyramidal cells with a stack of levels are constructed using skip connections or shortcuts. According to Hinton et al. (2011), traditional ResNet representation uses dual or three- layer shortcuts to incorporate the nonlinearity representations with the ReLU layer. Additionally, ReLU is used to perform batch normalization between nonlinearity representations. The learning process of the skip weights eliminates superfluous weight matrices to reduce layer structure and stacking.

Aim

With the aid of the earlier layer's activation function and the subsequent layers' learning weights, the problems associated with vanishing gradients were avoided (Farouk, A. et al., 2015). According to Graves et al. (2006), during the training process, the surrounding layers' learnt weights get used to the up layer's quietness and enhance the previously skipped layer. Layer skipping, which makes the network simpler during earlier training stages, is novel in the RNN.

Objective

The RNN network gradually replaces the skipped levels as it learns feature space mapping. The learning phase is carried out more quickly because the mapped layer of the feature space in the training set is enlarged to multiple (Chen et al., 2014). More learning phases of the feature space are discovered by deep networks with residual parts (Wang, M. M., et al., 2015). The



RNN models' learning and training phases expose it to more case studies and necessitate. Therefore, in this paper, we used a Capsule network (CapsNets) long with the RNN (resNet) type models to avoid such discrepancies.

Implementation

Speech is the fundamental form of communication between the speaker and the listener. Machines are unable to understand the speaker's words, while humans can easily understand their meaning. Today, machines take up the majority of our lives, but we are unable to communicate with them. Speech recognition requires both the human brain and machine learning technology to interface with humans and robots. This paper uses English as the speech recognition language because it is a universal language. The feature learning capability is employed in many assignments to implement the machine learning technique.

Speech Recognition with RNN

The input information grouping for RN networks is determined by the constant "a," where $a = (a_1, \dots, a_Y)$; a common discontinuous RNN memory masked vector collection is "V." where v is equal to $(v_1, \dots, v_Y)$. The resulting vector sequence is shown as 'C', where $C = (c_1, \dots, c_Y)$, with the related condition from $t = 1$ to T being highlighted. The 'X' expressions in equation 1 denote weight grids. The information-obscured weight network is called X_{av} . The 'd' condition denotes average predisposition vectors, and 'dv' stands for masked inclination vector and 'V' for hidden layer exertion.

$$V_x = X(X_{av}C_t + X_{vv}V_{t-1} + d_v) \quad (1)$$

$$C_t = X_{vv}V_{t-1} + d_c \quad (2)$$

'V' is a fundamental component of a sigmoid capacity function. Additionally, the generated building blocks are stored in the memory cells thanks to the Bidirectional Long Short-Term Memory that we recently discovered. In this paper, a bidirectional long short-term memory has been used. The creation of the BiLSTM memory cells was primarily driven by the need to store data. These memory blocks have the advantage of being better at locating and utilising long-range settings than LSTM cells, which is why BiLSTM memory cells were developed in place of LSTM.

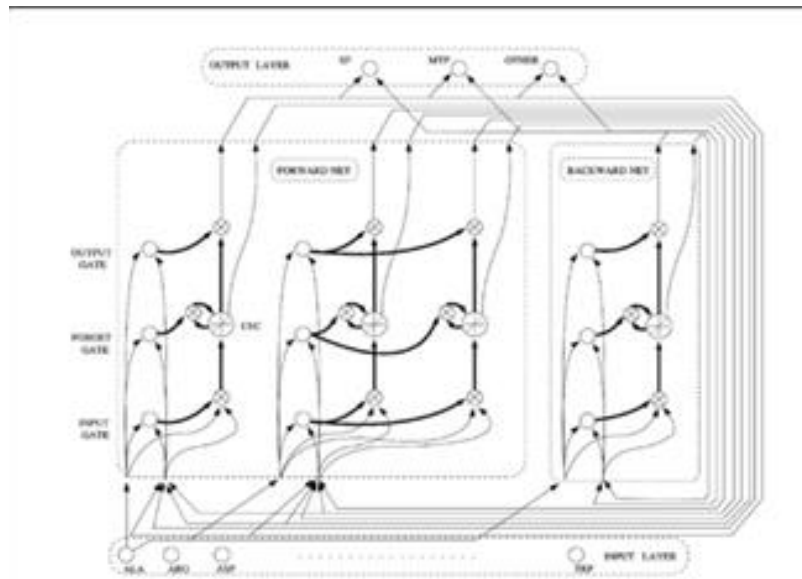


Figure 1: Bilstm Memory Cell

Fig.1 shows an introverted bilstm memory cell. The interpretation of bilstm will make the 'v' is implemented through the supplementary combination capacity by the memory cells.

$$j_t = \mu (X_{aj}c_t + X_{vj}v_{t-1} + X_{bj}v_{b-1} + d_j) \quad (3)$$

$$g_t = \mu (X_{ag}c_t + X_{vg}v_{t-1} + X_{bg}v_{b-1} + d_g) \quad (4)$$

$$c_t = g_t b_{t-1} + j_t \tanh (X_{ac}c_t + X_{vc}v_{t-1} + X_{bc}v_{b-1} + d_c) \quad (5)$$

$$i_t = \mu (X_{ai}c_t + X_{vi}v_{t-1} + X_{bi}v_{b-1} + d_i) \quad (6)$$

$$v_t = i_t \tanh (c_t) \quad (7)$$

The voice characteristic extractor using a recurrent neural network is then developed for a variety of speech features. The recurrent neural network then classifies the stolen speech properties as phonemes. Where "" denotes the logistic sigmoid function and the input gate, forget gate, output gate, and cell activation vectors, respectively, are

All the same size as the hidden vector "v," and the

Logistic sigmoid function is represented by "j," "g," "e," and "i." the weight matrices are the cell to gate vectors, and "x" is a diagonal vector. Therefore, the input element 'x' of the cell vector serves as the source for the output building blocks 'y' in

Each gate vector, which in turn receive the input data at time instances from t to t+1 and so forth.

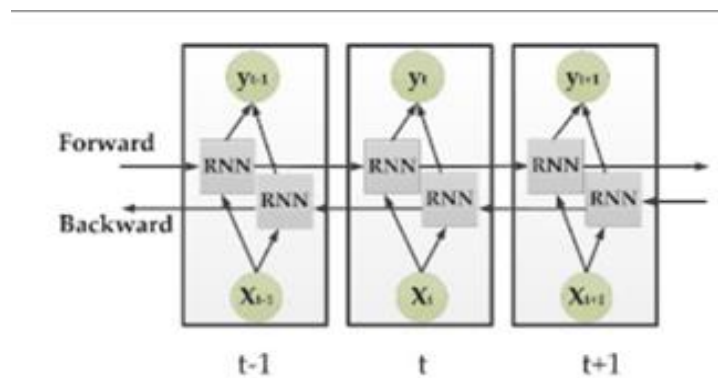


Figure 2: *Bidirectional RNN*

The Bidirectional RNN Speech Recognition Model Network is shown in Figure 2. Expressions in Data are transliterated only once. Furthermore, there is no justification to make use of any potential context. By distributing the speech recognition data along with the routes over two hidden layers, bidirectional RNNs achieve this. The two distinct hidden layers then flow through to the same output layer. However, in traditional RNNs, the data are gathered from the prior context-related pieces of information. The RNN layer in our suggested architecture employs a 1×1 convolution layer. This convolution layer architecture is used to minimise the number of parameters for weights and biases. Parameters will increase the depth of the structure by reducing the weights and biases.

By swapping each hidden sequence x, v through the forward and backward progressions x, v and x, y , the bidirectional RNNs can be realised. Next, create a convincing hidden layer that receives input from both the forward and backward layers equally. For the hidden layers, the LSTM structure is used, but more training rounds are required to employ a deep bidirectional LSTM structure. The deep Bi-LSTM has been used for voice recognition for the first time and, because of its structure, accepts a stunning improvement over the single-layer LSTM and traditional RNNs, where the data are gathered from the prior context pieces of information.

3.2 Speech Recognition with Capsule Net

A paradigm known as a "capsule scheme" converts a scalar yield neuron into a vector yield neuron called a "capsule." By utilising vector neurons, the capsule network is able to store a variety of pose data with small vector sizes, such as object translation and rotation.

The capsule architecture is first intended to instruct items regarding their length and posture data. The capsule network can train the hierarchical connection among these pose data-controlled capsules. The routing method is now carried out by concurrence among the capsules. The obtained scalar neuron values from the RNN are then replaced with those from the capsule network. The scalar neuron values are the neural networks that Recurrent Neural Networks (RNNs) account for most frequently. These recurrent neural networks connect to the capsule-shaped vector neuron. The capsules' r in this structure range from 1 to ' r_s, N '. These variables indicate the i th capsule of the N th layer and the r th capsule of the $(N + 1)$ th layer, respectively. Finally, $r(N+1)$ describes the values of layers and capsules generated from ' r_s, N '.



$\exp(XN_i * C_{in}) / M_{jnx} (13), r (N+1),$

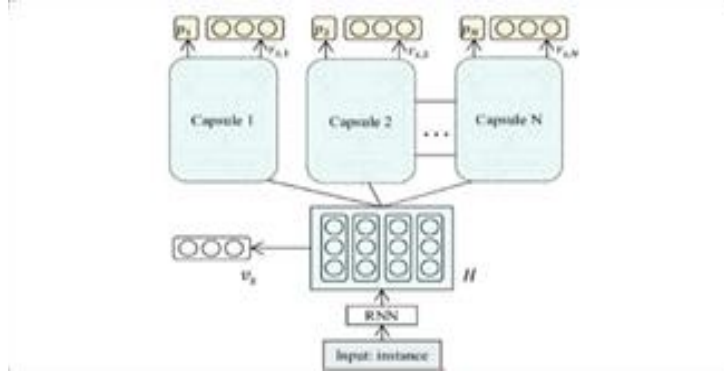


Figure 3: illustrates speech recognition with the capsule net

model. M_{jn} is a weight matrix among the capsules r and $r (N+1) j$, and C_{jN} is a coupling coefficient. We could compute c_{jn} by applying softmax function to b_{ij} as

$$C_{jn} = P \exp(d_{jn}) / \sum_k \exp(d_{jk}) \quad (14)$$

d_{jk} corresponds to the log prior probability that capsule j in layer N must be joined to the k capsule to the layer of $(N + 1)$; in addition to that, these functions are learned through the routing algorithm.

The output capsule values of the capsnet values are then changed via the a_{jn} once more, and this process is repeated for 't' routing times. By using the dot product of $W(j)$ and $W(N+1)j$ in equation 15, agreement-based routing may represent these 't' routing periods. The coupling coefficient updates the results and the calculated concurrence between the child and parent capsules. With the aid of the squashing function, the networks are capturing functions for train. According to the squash functions, short vectors have a short measurement length close to '0' while long vectors are close to '1'.

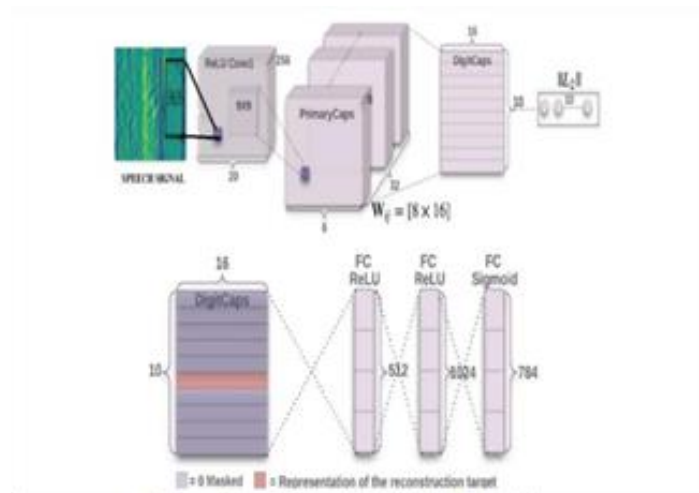


Figure 4: Speech Recognition RNN with Capsule Net

This speech recognition RNN with capsule net layers are necessary through the training period



of the RNN. This classifier layer contains a 9*9 average pooling layer, a stride of 3, and a convol of 1*1 along 128 filters. The global average pooling layers of 1024 outputs and 784 outputs and softmax classification layers are attained. The uses of these layers are reducing the combating gradient vanishing problem as well offer regularization. The whole inception module structure consists of 32 layers deep. This architecture was intended to develop to maintain calculation efficiency as well as low computational resources.

The voice data are trained using the temporal information of the speech at the RNN Capsules (RNCaps). The critical bits of information by temporal information of cues meant for emotion recognition are present in the speech data systematically, the development.

3 Results

The high-level features of the voice signal are obtained using trained RNN layer-based feature extraction. The new speech feature for the capsule network is generated from the generated high-level features. Final step: Using the fully connected network, the obtained speech features and attribute features are combined into the same RNN with Caps Net framework. The suggested architecture is trained using a variety of dimension of kernels in comparison to the baseline RNN models. Additionally to these channels, the fully connected neural network layers yield promising results. Then, the layers of the capsule structure network are created in order to create the capsnet layer. The best agitated constraints in a sequence of classifying the kernel sizes followed the training and testing method.

The most effective agitated constraints in a series for classifying kernel size after training and testing are 10. And the size channel utilised for testing and training is 12.0 M by 82 in terms of the number of parameters used. The optimal 8 input vectors and 16 output vector lengths determine the appropriate sequence length.

Table 1 The output of different models

Model	Number of Parameters	Clean Error	Noisy Error
RNN single layer 1	12.0 M	30.1 %	59.2 %
RNN single layer 2	8.0 M	28.5%	58.2 %
RNN single layer 3	82.98M	26.2%	60.3%
Caps kernel 9	122.3 M	23.4 %	59.4 %
Caps kernel 15	118.0 M	22.3 %	54.2 %
Caps kernel 20	93.5 M	21.1%	59.5 %
Caps kernel 28	148.6 M	20.7 %	59.3%
Caps No Decoder	72.1 M	14.8 %	56.4 %
Caps channel 82	148.0 M	13.4 %	57.7%
Caps output vec 8	14.9 M	10.7 %	45.2 %
Caps output vec 16	12.9 M	10.2 %	48.7 %

The first input kernel size in this experiment is 9. The clean and noisy errors are determined by enlarging the kernel and increasing the number of parameters.

The number of parameters utilised in kernels is initially reduced till kernels 20. However, if



the kernel size is higher than kernel 20, the speech information after the next convolutional layer becomes uninteresting. To create the voice information dimension positive along with the regular convolutional layer, additional zero-padding is added.

4 Conclusion

Speech is the fundamental form of communication between the speaker and the listener. Machines are unable to understand the speaker's words, while humans can easily understand their meaning. Machines now take up a large portion of our lives, but machine interaction is not possible. The human brain, like machine learning technology, is necessary for voice recognition in order for humans to engage with machines. This paper uses English as the speech recognition language because it is a universal language. Through the feature learning capability, the machine learning methodology is utilised in many assignments. The results of the data modelling capabilities are superior to those of the conventional learning methodology. As a result, a machine-learning technique is used in this work to combine the voice features and attributes for speech signal detection.

References

1. Basha, S; Dubey; S.R; Pulabaigari; V. & Mukherjee S, Year:2020, "Impact of fully connected layers on performance of convolutional neural networks for image classification", Neurocomputing. 378, pp. 112–119.
2. Castro; J.L; Delgado M; Year: 1996, Fuzzy systems with defuzzification are universal approximators. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1), pp.149-152.
3. Chen; Guoguo; Carolina Parada; Georg Heigold, Year:2014, "Small-footprint keyword spotting using deep neural networks." In IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 4087-4091.
4. P Ghahremani; Baba Ali B; Povey D; Riedhammer K; Trmal J; Khudanpur S, Year: 2014, "A pitch extraction algorithm tuned for automatic speech recognition", Glorot X; Bengio Y, Year: 2010.
5. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249-256.