ORIGINAL RESEARCH PAPER

Volume-7 | Issue-9 | September-2018 | PRINT ISSN No 2277 - 8179

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH

SPEECH RECOGNITION USING CONVOLUTION NEURAL NETWORK

Engineering

Maisnam Niranjan Singh Research scholar, Dayanand Sagar Engineering(VTU)

Y.S Kumaraswamy* Prof., Dean R & D(Nagarjuna Engineering College) *Corresponding Author

ABSTRACT

Speech is a vital form of communication among humans and thus understanding speech in the context of communication results in directly influencing our actions. Researchers all around the world have been trying to teach machines how to recognize speech. Speech understanding and recognition by machines even though is a challenging proposition but nevertheless researchers have been successful in applying machine learning algorithms to speech recognition which is commendable. With the advent of deep learning researchers have further enhanced the state of the art of speech recognition. With more computation speed and GPU , it has become possible to train large datasets even in PC and Laptops.

In this paper we adopt a deep learning approach to teach machine how to convert speech into text. The main objective of this paper is to apply deep learning techniques to speech data and convert into textual form using Convolution neural network . It has always been believed in the deep learning community that CNN is good only with classifying images. But recent trends have suggests that CNN are applied to NLP , Speech and other domains for better results and have shown that it indeed works .

KEYWORDS

1.INTRODUCTION

Speech recognition techniques have evolved over the years. Initially, researchers have successfully applied Hidden Markov Models to speech recognition. But with the advent of deep learning many researchers have successfully used deep learning techniques such as recurrent neural network and convolution neural networks to classify speech.

Technical advancements have fueled the growth of speech interfaces through the availability of machine learning tools, resulting in more Internet-connected products that can listen and respond to us than ever before. Recently Mozilla released it's speech recognition software to the public. We believe this technology can and will enable a wave of innovative products and services, and that it should be available to everyone.

And yet, while this technology is still maturing, we're seeing there a lot of unknown areas to be explored and remove the barriers which would improve the speech recognition technology.

Recently Mozilla also made publicly available voice dataset, which was contributed to by nearly 20,000 people globally.

This paper, has taken the CNN route rather than sequence modelling neural network like recurrent neural network, LSTM or the recent published GTU. We provide the architecture of CNN and we also provide code snippets for both the model build phase and model training phase an the accuracy and error logs to see how CNN can also be applied to speech data rather than images , but due to other constraints we do not provide the full source code rather we provide only a few code to get the idea.

To help predict text from speech we chose to collect data from the internet and other sources .and clean up the data so that these sentences can be fed into the neural network layer. We used Scikit-learn, a Python machine learning framework along with Pandas, Numpy, Keras, Tensorflow and CNN .

With this we believe that some of the applications such as movie sub titles could be generated by listening to the movie dialogues. The list of applications for speech recognition are many , imagine a person speaking in his or her native language could be translated into text , this would provide a giant step in today's communication.

We choose Convolution neural network , a form of neural network used mostly in classification of images to identify speech recognition and emits words from the audio signals.

2. Previous Work

Li Deng and Xiao Li [9] paved the way for many Machine learning paradigms that are motivated in the context of ASR technology and applications. These new insights from modern ML methodology shows great promise to advance the state-of-the-art in ASR technology. They intended to foster further cross-pollination between the ML and ASR communities than has occurred in the past.

3.Data Types

TensorFlow recently released the Speech Commands Datasets. It includes 65,000 one-second long utterances of 30 short words, By improving the recognition accuracy of open-sourced voice interface tools, we can improve product effectiveness and their accessibility.

One reason so few services are commercially available is a lack of data. Startups, researchers or anyone else who wants to build voice-enabled technologies need high quality, transcribed voice data on which to train machine learning algorithms. Right now, they can only access fairly limited data sets.

To address this barrier, we launched Project Common Voice this past July. Our aim is to make it easy for people to donate their voices to a publicly available database, and in doing so build a voice dataset that everyone can use to train new voice-enabled applications.

Today, we've released the first tranche of donated voices: nearly 400,000 recordings, representing 500 hours of speech. Anyone can download this data.

4. Process flow

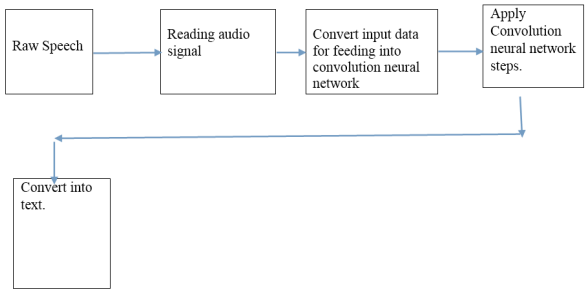

FIGURE 1

The above figure 1 shows the flow diagram of the Speech recognition architecture. The dataset is divided into training an testing sets. Further from the training set a validation is reserved to serve as a virtual way of forming the actual testing set. The data size is not much and since the data size is less there was always a chance of overfitting , and the avoid

overfitting we could have made synthetic audio noise data as part of data augmentation technique. Further, the convolution layers convolution and applying activation function to the result along with the bias and then pooling could have been added to see if it improved the accuracy.

5.Speech CNN implementation .

```
window_size=.02
window_stride=.01
window_type='hamming'
normalize=True
max_len=101
batch_size = 64
train_iterator = SpeechDirectoryIterator(directory=train_path,
batch_size=batch_size,
                                         window_size=window_size,
                                        window_stride=window_stride,
                                         window_type=window_type,
                                         normalize=normalize,
                                         max_len=max_len)
```

FIGURE 2

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,
Dropout
model = Sequential()
model.add(Conv2D(12, (5, 5), activation = 'relu', input_shape=train
iterator.image_shape))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(25, (5, 5), activation = 'relu'))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(180, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(100, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(len(classnames), activation = 'softmax')) #Last
layer with one output per class

model.compile(loss='categorical_crossentropy', optimizer='Adam',
metrics=["accuracy"])
model.summary()
```

FIGURE 3

The above figure 3 shows the code for ingesting audio files using custom made similar to Keras function Directory Iterator to take data using batch size.

6. Architecture

In order to perform speech recognition , the raw audio file which contains short speeches , which are in the form of commands are read using librosa a python library to read audio files. As the speech length are of different lengths , we need to convert these audio files into same size.

Step 1	Step 2	Step 3	Step 4	Step 5
Speech data	Read data	Augmentation	CNN	Prediction

FIGURE 4

6. CNN parameters.

We will be showing only the important part of the code , the CNN architecture summary is shown below, the model architecture and the parameters which is 3,690,329 .

The model summary.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 97, 157, 12)	312
max_pooling2d_1 (MaxPooling2	(None, 48, 78, 12)	0
conv2d_2 (Conv2D)	(None, 44, 74, 25)	7525
max_pooling2d_2 (MaxPooling2	(None, 22, 37, 25)	0
flatten_1 (Flatten)	(None, 20350)	0
dense_1 (Dense)	(None, 180)	3663180
dropout_1 (Dropout)	(None, 180)	0
dense_2 (Dense)	(None, 100)	18100
dropout_2 (Dropout)	(None, 100)	0
dense_3 (Dense)	(None, 12)	1212

Total params: 3,690,329
Trainable params: 3,690,329
Non-trainable params: 0

```
from keras.callbacks import EarlyStopping, ReduceLROnPlateau

early = EarlyStopping(monitor='val_loss', min_delta=0, patience=5,
verbose=1, mode='auto')

reduce = ReduceLROnPlateau(monitor='val_loss', factor=0.5,
patience=3, verbose=1, mode='auto', min_lr=0.00001)

model.fit_generator(train_iterator,

      steps_per_epoch=int(np.ceil(train_iterator.n / batch_size)),
      epochs=8,

      validation_data=val_iterator,

      validation_steps=int(np.ceil(val_iterator.n / batch_size)),

      verbose=1, callbacks=[early, reduce])
```

```
Epoch 1/8
338/338 [==============================] - 418s 1s/step
- loss: 1.5899 - acc: 0.4472 - val_loss: 0.8523 - val_acc: 0.7465

Epoch 2/8
338/338 [==============================] - 393s 1s/step
- loss: 0.8568 - acc: 0.7113 - val_loss: 0.6969 - val_acc: 0.7875

Epoch 3/8
338/338 [==============================] - 426s 1s/step
- loss: 0.6566 - acc: 0.7807 - val_loss: 0.5926 - val_acc: 0.8170

Epoch 4/8
338/338 [==============================] - 373s 1s/step
- loss: 0.5111 - acc: 0.8309 - val_loss: 0.5835 - val_acc: 0.8306

Epoch 5/8
338/338 [==============================] - 388s 1s/step
- loss: 0.4342 - acc: 0.8584 - val_loss: 0.5437 - val_acc: 0.8359

Epoch 6/8
338/338 [==============================] - 382s 1s/step
- loss: 0.3597 - acc: 0.8818 - val_loss: 0.5694 - val_acc: 0.8413

Epoch 7/8
338/338 [==============================] - 379s 1s/step
- loss: 0.3074 - acc: 0.8997 - val_loss: 0.5793 - val_acc: 0.8483
```

```
Epoch 8/8
338/338 [==============================] - 381s 1s/step
- loss: 0.2644 - acc: 0.9172 - val_loss: 0.6259 - val_acc: 0.8394
<keras.callbacks.History at 0x2440cea3438>
```

FIGURE 5

As we can see after running 8 epochs we achieve a score of 0.9172 on the training data and a score of 0.8394 on validation which indicates overfitting.

6. Training Results and Discussion

The training result in Figure 5 shows that it achieved 0.9172 while validation score is merely 0.8394 which means it has overfitted . To alleviate overfitting we could have added artificially generated noise and could have trained along with the data set. We haven't used BatchNormalisation technique and it would be interesting to see how batch normalization would help to the overall performance.

1. Prediction Results and Discussion

```
import math
from keras.utils import Sequence
from keras.preprocessing.image import img_to_array

def loadAndSpect(fname, window_size, window_stride,
window_type, normalize, max_len):
   img = spect_loader(os.path.join(test_path_audio, fname),

                      window_size,
                      window_stride,
                      window_type,
                      normalize,
                      max_len)
   img=np.swapaxes(img, 0, 2)

   x = img_to_array(img, data_format='channels_last')
   return x

class WavSequence(Sequence):

def __init__(self, x_set, batch_size=64, window_size=0.02,
window_stride=0.01, window_type='hamming',

     normalize=True, max_len=101):

       self.x = x_set
       self.batch_size = batch_size
       self.window_size = window_size
       self.window_stride = window_stride
       self.window_type = window_type
       self.normalize = normalize
       self.max_len = max_len

def __len__(self):
return math.ceil(len(self.x) / self.batch_size)

def __getitem__(self, idx):
batch_x = self.x[idx * self.batch_size:(idx + 1) * self.batch_size]

return np.array([
loadAndSpect(file_name, window_size, window_stride,
window_type, normalize, max_len)
            for file_name in batch_x])

seq = WavSequence(test_filenames, batch_size=batch_size)
preds = model.predict_generator(generator=seq, steps=len(seq),
workers=1,verbose=1)

2478/2478 [==============================] - 2714s
1s/step
```

FIGURE 6

7. Performance improvement using different architecture

We believe that trying with sequence modelling could have been used instead of CNN. The reason why we chose CNN is that the data is not a long conversation data rather a sample of short form of speech is uttered in a few seconds. While doing we could have added noise data which are generated artificially to the datasets , these noise data could have certainly improve the accuracy of the models.

To improve performances different deep learning architecture could be used, hyper parameters could be selected and selected and tuned. Different optimizers could be used and the learning rate could be adjusted and different iterations could be experimented. We believe that using different architecture, using dropouts to avoid overfitting, using Batch Normalization and tuning of hyper parameters could help boost the performance of sentiment analysis.

8. Conclusion and Future Work

Teaching machines speech recognition is extremely challenging task and thus there is ample scope for improvement. Even using deep learning needs a lot of improvement. Speech recognition is indeed a very interesting topic and it is still very limited. Our aim and objective as mentioned before is to give some reference point from where researchers and other deep learning students can get an idea how to start processing audio files and convert into text.

We hope that this paper has brought some ideas to try out and get good accuracy and hope it has found to be interesting and useful.

We hope this paper brings about understanding and inspiration amongst the research communities of ASR.

REFERENCES

[1] Astrahan, M., "Speech Analysis by Clustering or the Hyperphoneme Method," Al Memo 124, Computer Science Department, Stanford University, Stanford, California (1970).
[2] Bobrow, D.G. and D.H. Klatt, "A Limited Speech RecognitionSystem," Proc. FJCC (19G8) 305-318.
[3] Chomsky, N. and M.Halle, "The Sound Pattern of English," Harper and Row, New York (1968).
[4] Fant, G., "Acoustic Theory of Speech Production," Mouton and Company: The Hague (19G0).
[5] Flanagan, J.L., "Speech Analysis, Synthesis, and Percept ion," Academic Press: New York (19G5).
[6] Gold, B., "Word Recognition Computer Program," RLE Report No. 452, MIT, Cambridge, Mass. (19GG).Prospects SPEECH WORKING PAPERS 5-14
[7] Hyde. S.R.. "Automatic Speech Recognition: Literature Survey and Discussion," RDR No. 45, Post Office Research Department, Doll is Hi I I, London N.W.2 (19G8).
[8] Kozhevnikov, V.A. and L.A.Chistovich, "Speech: Articulation and Perception," Moscow-Leningrad (1965). Translated by Joint Publication Research Service, Washington, D. C.
[9] Lehiste, I., "Readings in Acoustic-Phonetics," MIT Press, Cambridge, Mass. (1967).
[10] Londgren, N., "Machine Recogni t ion of Human Language," IEEE Spectrum 2, Nos. 3 and 4 (1965).
[11] Newell, A., "Heuristic Programming: III Structured Problems," in J. S. Avonofsky (ed.), Progress in Operations Research, Vol 3 (John Wiley and Sons) 363-415.
[12] Newell, A., J.Barnett, J.Forgie, C.Green, D.KIatt, J.C.R.Licklider, J.Munson, R.Reddy, and W.Woods, "Final Report of a Study Group on Speech Understanding Systems," Comp. Sci. Dept., Carnegie-Mel Ion Univ., Pittsburgh, (1971).
[13] Pierce, J.R., "Whither Speech Recognition," J. Acoust. Soc. Am. 46 (1966) 1849-1051.
[14] Reddy, D. R., "Computer Recognition of Connected Speech," J. Acoust. Soc. Am. 42 (1967) 329-347.
[15] Reddy, D.R. and P.J.Vicens, "A Procedure for Segmentation of Connected Speech," J. Audio Eng. S o c , 16,4 (1968) 404-412.
[16] Saki, T. and S.Doshita, "The Automatic Speech Recognition System for Conversational Sound," IEEE Trans, on Electronic Computers, 12 (1963) 835.
[17] Stevens, K.N. and M.Halle, "Speech Recognition: A Model and a Program for Research," IRE Trans. PGIT, IT-8 (1962) 155-159.
[18] Tappert, C.C., N.R.Dixon, D.H.Beetle, and W.D.Chapman, "The Use of Dynamic Segments in the Automatic Recognition of Continuous Speech," IBM Corp., RADC-TR-70-22, Rome Air Development Center,Rome, New York (1970).
[19] Vicens, P.J., "Aspects of Speech Recognition by Computer," Pn.D. Thesis, Computer Science Department (Report No. 127), Stanford University, California (1969).
[20]. Sadaoki Furui, 50 years of Progress in speech and Speaker Recognition Research , ECTI Transactions on Computer and Information Technology,Vol.1. No.2 November 2005.
[21]. K.H.Davis, R.Biddulph, and S.Balashek, Automatic Recognition of spoken Digits, J.Acoust.Soc.Am., 24(6):637-642,1952.
[22]. H.F.Olson and H.Belar, Phonetic Typewriter , J.Acoust.Soc.Am.,28(6):1072-1081,1956.
[23]. D.B.Fry, Theoritical Aspects of Mechanical speech Recognition , and P.Denes, The design and Operation of the Mechanical Speech Recognizer at Universtiy College London, J.British Inst. Radio Engr., 19:4,211-299,1959.
[24]. J.W.Forgie and C.D.Forgie, Results obtained from a vowel recognition computer program , J.A.S.A., 31(11),pp.1480-1489.1959.
[25]. J.Suzuki and K.Nakata, Recognition of Japanese Vowels Preliminary to the Recognition of Speech , J.Radio Res.Lab37(8):193-212,1961.
[26]. T.Sakai and S.Doshita, The phonetic typewriter, information processing 1962 , Proc.IFIP Congress, 1962.
[27]. K.Nagata, Y.Kato, and S.Chiba, Spoken Digit Recognizer for Japanese Language , NEC Res.Develop., No.6,1963.
[28]. T.B.Martin, A.L.Nelson, and H.J.Zadell, Speech Recognition b Feature Abstraction Techniques , Tech.Report AL-TDR-64-176,Air Force Avionics Lab,1964.
[29]. T.K.Vintsyuk, Speech Discrimination by Dynamic Programming , Kibernetika, 4(2):81-88,Jan.-Feb.1968.
[30]. H.Sakoe and S.Chiba, Dynamic programming algorithm optimization for spoken word recognition ,IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-26(1).pp.43- 49,1978.
[31]. D.R.Reddy, An Approach to Computer Speech Recognition by Direct Analysis of the Speech Wave , Tech.Report No.C549, Computer Science Dept., Stanford Univ., September 1966.
[32]. V.M.Velichko and N.G.Zagoruyko, Automatic Recognition of 200 words , Int.J.Man-Machine Studies,2:223,June 1970.
[33]. H.Sakoe and S.Chiba, Dynamic Programming Algorithm Optimization for Spoken

Word Recognition ,IEEE Trans.Acoustics, Speech, Signal Proc.,ASSP-26(1):43-49,February 1978.
[34]. F.Itakura, Minimum Prediction Residula Applied to Speech Recognition ,IEEE Trans.Acoustics, Speech,Signal Proc., ASSP-23(1):67-72,February 1975.
[35]. C.C.Tappert,N.R.Dixon, A.S.Rabinowitz, and W.D.Chapman, Automatic Recognition of Continuous Speech Utilizing Dynamic Segmentation, Dual Classification, Sequential Decoding and Error Recover , Rome Air Dev.Cen, Rome, NY,Tech.Report TR-71-146,1971.
[36]. F.Jelinek, L.R.Bahl, and R.L.Mercer, Design of a Linguisistic Statistical Decoder for the Recognition of Continuous Speech , IEEE Trans.Information Theory,IT- 21:250-256,1975.
[37]. F.Jelinek, The Development of an Experimental Discrete Dictation Recognizer , Proc.IEEE,73(11):1616- 624,1985.
[38]. L.R.Rabiner, S.E.Levinson, A.E.Rosenberg, and J.G.Wilpon, Speaker Independent Recognition of Isolated Words Using Clustering Techniques , IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-27:336- 349,August 1979.
[39]. D.Klatt, Review of the ARPA Speech understanding project , J.A.S.A. 62(6),pp.1324-1366,1977.
[40]. B.Lowrre, The HARPY speech understanding system , Trends in Speech Recognition, W.Lea,Ed., Speech Science Pub., pp.576-586,1990.
[41]. H.Sakoe, Two Level DP Matching A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition , IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-27:588-595, December 1979.
[42]. J.S.Bridle and M.D.Brown, Connected Word Recognition Using whole word templates , Proc. Inst.Acoust.Autumn Conf.,25-28,November 1979.
[43]. C.S.Myers and L.R.Rabiner, A Level Building Dynaimc Time Warping Algorithm for Connected Word Recognition , IEEE Trans. Acoustics, ASSP-29:284-297,April 1981.
[44]. C.H.Lee and L.R.Rabiner, A Frame Synchronous Network Search Algorithm for Connected Word Recognition , IEEE Trans. Acoustics, Speech, Signal Proc., 37(11):1649-1658, November 1989.
[45]. J.Ferguson, Ed., Hidden Markov Models for Speech, IDA,Princeton, NJ,1980.
[46]. L.R.Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition , Proc.IEEE,77(2):257-286,February 1989.
[47]. J.Ferguson, Ed., Hidden Markov models for speech, IDA, Princeton, NJ,1980.
[48]. L.R.Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition , Proc.IEEE,77(2),pp.257-286,1989.
[49]. L.R.Rabiner and B.H.Juang, Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliff, New Jersy, 1993.
[50]. S.Furui, Speaker-independent isolated word recognition using dynamic features of speech spectrum , IEEE Trans. Acoustics, Speech, Signal Processing, ASSP- 34,pp.52-59,1986.
[51]. F.Jelinek, The development of an experimental discrete dictation recognizer , Proc. IEEE,73(11),pp.1616- 1624,1985.
[52]. S.Katagiri, Speech pattern recognition using neural networks , W.Chou and B.H.Juang (Eds.) Pattern Recognition in Speech and Language Processing, CRC Press, pp.115-147,2003.
[53]. R.P.Lippmann, An introduction to computing with neural nets , IEEE ASSP Mag., 4(2),pp.4-22,April 1987.
[54]. A.Weibel, et.al., Phoneme recognition using time-delay neural networks , IEEE Trans. Acoustics, Speech, Signal Proc.,37,pp.393-404,1989.
[55]. K.F.Lee, et.al, An overview of the SPHINX speech recognition system , Proc.ICASSP, 38,pp.600-610,1990.
[56]. Y.L.Chow, et.al. BYBLOS, the BBN continuous speech recognition system , Proc.ICASSP, pp.89-92,1987.
[57]. M.Weintraub et al., Linguistic constraints in hidden Markov model based speech recognition , Proc.ICASSP, pp.699-702,1989.
[58]. D.B.Paul, The Lincoln robust continuous speech recognizer , Proc.ICASSP,449-452,1989. [40].V.Zue, et.al., The MIT summit speech recognition system, a progress report , Proc.DARPA Speech and Natural Language Workshop, pp.179-189,1989.
[59]. C.H.Lee, etc.al., Acoustic modeling for large vocabulary speech rcognition ,Computer Speech and Language, 4,pp.127-165,1990.
[60]. B.H.Juang and S.Furui, Automatic speech recognition and understanding: A first step toward natural human machine communication , Proc.IEEE,88,8,pp.1142- 1165,2000.
[61]. K.P.Li and G.W.Hughes, Talker differences as they appear in correlation matrices of continuous speech spectra , J.A.S.A., 55,pp.833837,1974.
[62]. C. J. Leggetter and P. C. Woodland, Maximum likelihood linear regression for speaker pp.adaptation of continuous density hidden Markov models, Computer Speech and Language, 9, 171-185, 1995.
[63]. A. P. Varga and R. K. Moore, Hidden Markov model decomposition of speech and noise, Proc. ICASSP, pp.