APPROACH OF DIFFERENT MODELS OF MACHINE LEARNING IN AUTOMATIC SPEECH RECOGNITION OF BALKAN LANGUAGES

Dejan Dodić
EDUKOM d.o.o. Vranje, Republic of Serbia, dodic@edukom.rs

Abstract: Over the last few decades, there has been a tremendous development of machine learning paradigms used in automatic speech recognition (ASR) to automate the home (smart home) to space exploration. Although commercial speech recognizers are available for certain well-defined applications such as dictation and transcription, with a special emphasis on e-Dictate applications for Balkan language recognition, many problems in ASR such as recognition in noisy environments, multilingual recognition and multimodal recognition have yet to be addressed efficiently. A comprehensive overview of common machine learning (ML) techniques such as artificial neural networks, vector support machines, and Gaussian mixture models is provided along with the hidden Markov models represented in ASR.
Keywords:Automatic speech recognition (ASR), Gaussian mixture models, Hidden Markov models, Machine learning, Deep learning and speech, JAVA programming, Speech recognition (SR), e-Dictate

1. INTRODUCTION

From the early part of the previous century, there was a curiosity about how computers could do what only humans could perceive, such as speech recognition, understanding natural language, image processing, and so on. Speech is the primary, most effective way of communication between people, speech recognition research has aroused a lot of enthusiasm over the last five decades since the advent of artificial intelligence. Many reasons can be attributed to this enthusiasm, from mere technological curiosity, to the desire to automate tasks using machines.
The study of speech analysis dates back to the early nineteenth century, when Homer Dudley of the Bell Laboratory made the first proposal for a speech analysis and synthesis system in the 1930s.
A significant achievement occurred in 1959 when a phoneme recognizer was developed that recognizes four vowels and nine consonants using statistical information on phoneme sequences in English. This marked the first use of statistical syntax in speech recognition.
A valuable technique that became popular in the 1970s was dynamic automatic speech recognition (ASR) programming. This was to meet the goal of providing telecommunications services to people, including voice dialing and command-based automation of telephone calls.
Voice is considered an important biometric, and multimodal recognition systems are emerging to enhance the robustness of authenticity. Speech recognition can be extended to speaker recognition, using information present in speech and other various methods.

2. OBJECTIVE OF THE RESEARCH

Over the years, sophisticated skills for recognizing patterns such as speech, handwriting, facial features, etc. have been developed. The tendency towards computer programs that learn computers from previous experience gives birth to machine learning. Mitchell stated in the context of machine learning that: "A computer program is said to learn from experience E, given a class of tasks in T and a measure of P performance, if its performance on tasks in T, as measured by P, improves experience E. "
In supervised learning, the machine is trained with a marked data set where the output response or class is known for each input data vector. The assumption is that, if the training data are large enough, a hypothesis can be obtained that can have a good effect on the test data.
A simple example of supervised learning is the problem of curvature. Given the set of input data, the machine is capable of generating a curved surface that best fits the training data set, and during testing, the machine is expected to properly interpolate new data across the curved surface. Reverse neural networks such as perceptrons (adoption of delta learning rules or perceptron learning rules), multilayer perceptrons (MLP, feedback propagation adoption), and limited MLP fall into this category.
In unsupervised learning, the machine is expected to learn patterns in unmarked input data that it has set itself without any feedback from the environment. The problem can be stated as finding patterns in the input data set in order to appropriately divide or group the training data into subsets. Taxonomic problems, where devising efficient ways to group data into significant clusters, fall into this category. Examples are the Hebb and Hopfield networks, the Kohonen networks / self-organizing maps, and the adaptive resonance theory (ART) / ART (competitive learning) networks.

Automatic coding is a simple network that is trained to produce what is given at the input, ie. by setting the target output as the input. The network is trained to reproduce input data by the method of unsupervised backward gradient propagation.
In semi-controlled learning, unmarked data is also used to train the system. Typically, a small portion of tagged data with a large amount of unlabeled data is used. This approach to learning is usually adopted in problems where obtaining tagged data is very expensive.
In active learning, the algorithm interactively asks the user to get sample tags. This is used in scenarios where unmarked data is abundant, but tagging data is expensive.

3. RESEARCH METHODOLOGY

In this paper, we will focus on the results of experimental research of machine learning (ML) that translates spoken words into text in Balkan languages - Speech Recognation (ASR), as a specific form of verbal communication, based on artificial neural networks (ANN) through a concrete example. e-Dictate application. The paper also describes the e-Dictate application algorithm, which is specifically based on feature input (sound) and tag output (corresponding letter), calculated by the algorithm for classifying speech data on words spoken in a normal way, which was specially created for this study. Part of this database was used to test the ANN in the case of the above mentioned application. The case of speaker-dependent recognition was tested, and the results showed 93.44% accuracy in the case of speech recognition and 99, 38% in the case of natural voice recognition. In the case of whisper recognition, when ANN was trained for normal speech, the whisper recognition score was 59.00%.
Deep speech recognition has enabled you to develop your own JAVA application by following a specific JAVA syntax and dictating code. The application will convert the spoken words into text and after saving the program, import the file into the compiler and execute the program.
While using this application, the user must run a program with a class name followed by a declaration of elements and methods. The user can import packages and libraries at any stage of his program. The speech recognition application makes the programming process even easier with its prepared statements and blocks. For example, the user does not have to dictate each individual basic component of the main method, but only has to say the word "main". The application will open the basic "main class", including parentheses and empty spaces for arguments. At the end of the program, the user can dictate the word "submit" which will save and compile the program.
Moreover, ML can and occasionally uses ASR as an extensive, realistic application to rigorously test the effectiveness of a given technique and encourage new problems arising from the essentially sequential and dynamic nature of speech. On the other hand, although ASR is commercially available for some applications, it is largely an unresolved issue - for almost all applications, ASR performance is not equal to human performance. A new insight into the modern methodology of ML shows a great promise for the advancement of modern ASR technology.

4. RESEARCH RESULTS

In machine learning, it is known that the conversion of sound into text is a problem of classification. To train an audio transcription application, ML developers enter feature-labeled data into their model. This data is called a training set. The characteristics (sound) are entered and the labels (corresponding letter) are output, calculated by the classification algorithm.
Experiments performed at Microsoft have shown that with an increase in the amount of training data in the range of close to four orders of magnitude (from TIMIT to voice search to control panel), DNN-based systems monotonically outperform GMM-based systems, not only in absolute but also and in relative percentages. This is the kind of accuracy improvement not seen in the history of ASR.
The simplest but most important example is the use of context-sensitive (CD) phones, as well as the DNN output layer, originally invented in Microsoft Research.
Google's Android has a variety of applications that convert speech to text, as well as various editors that allow users to write a computer program in a user environment. An example of a speech converter is Google Speech Recognizer, which is used in the Google Voice app.
The Android speech package contains one interface and five voices. The Android app package contains high-level classes that cover the entire Android app model. An app can be defined using one or more of Android's core application components. The components of activities and services are defined in this package. The activity can start other activities, including activities that live in a separate application. A service is a component of an application that provides a screen with which a user can communicate to perform an action. For example, a service may play music, connect to a network, or work with a browser without the user being aware of the work in progress.

After receiving some results from the Voice to Text Converter (Voice to Text Converter is a predefined and coded Google speech recognition program. The current application uses libraries and extensions to use this API.), The program will start searching for matching words in its JAVA library and web search. If any match is found, we will see the result on the screen.

In the following text, we will consider how JAVA works in this domain. The entire process is located in the MainActivity class. To execute the command, the application needs to complete 6 steps and switch from one method to another. The onCreate method is the step in which an adapter is created and if there was an adapter created in the last cycle, it will be deleted. The next stage is when the application checks whether the voice recognition is running and working properly.

MainActivity and JavaConstants are the two basic classes of the application in which we stored the JAVA programming constants (for word guessing) and the complete application flow process.

MainActivity contains the main variable, methods and objects. This uses data from the JavaConstants class to validate the received data from the voice recorder. Database Helper is the second class in the second package that connects to different libraries and forwards the result to the JavaConstants class.

An activity diagram is a simple diagram that shows a user's journey from the moment they launch an application to the moment they complete a complete successful cycle.

In Android, speech recognition without dialogue is possible using another method, ie. by applying RecognationListener and replacing all its callback methods. That way, speech could be recognized without going to Google's original speech recognition dialog. The disadvantage of Android speech recognition without dialogue in custom activity is that we would also have to replace the onRmsChanged callback method if we needed to display voice visualization during recording.

e-Dictate is an Android application based on the JAVA programming language that converts speech into text, saves text and sends it via e-mail, in the form of SMS messages, or other messaging applications, as well as social networks, you can create notes and much more where text input is required.

e-Dictate has a wide application, in a word, in all situations as a replacement for a smartphone or computer keyboard.

As for the UI application, it is a completely user friendly application. The application is adapted and optimized for 35 world languages, among other things,including all languages of the Balkan states.

The application can be treated as a stenographer for business purposes, for the preparation of documents, writing, ie. dictation of blogs, reminders and any long and short textual content.

It is interesting to note that the application can also be used for quick correspondence. The percentage of success (accuracy) of the converted speech into text is higher than 98%, which clearly speaks about the quality of the e-Dictate application.

As an example, e-Dictate can play a special role in the safety of traffic participants, because typing messages with this application is a thing of the past.

The number of characters that can be dictated is unlimited. The user enters the text with his voice.

Through the e-Dictate application, several important options are permeated that fully help the user both in everyday life and for business purposes. When launching this application, the user can start his dictation by clicking on the microphone icon and the text will be displayed on the screen of the smartphone.

It is interesting to note that the application can be sent in the background and therefore the application will not stop listening and printing the text dictated by the user.

What makes e-Dictate completely different from other applications is the Audio Dictation option, which means that if the user is not able to dictate the text at the same time, he can record an .mp3 file through the application and subsequently play the mentioned .mp3 file. command the application to listen to .mp3 and convert the voice to text.

Also, a feature that especially distinguishes this app is the Translator option. If the user needs to translate the dictated text, in our example from one of the Balkan languages to any language in the world, by clicking on the Translator icon, the application goes to the next screen where the original dictated text is displayed at the top of the screen, selecting the language to translate and by pressing the Translate button, the text will be automatically translated into the desired language. By clicking on the arrow in the lower right corner, the user can forward (send) the translation via email, text message, or some other application for exchanging text messages, ie. communication.

The e-Dictate application has the option of storing text in the internal memory of the smartphone with the .txt extension. The user can export the .txt file for further text manipulation in one of the text editors of his smartphone or computer.

How does the backend e-Dictate application work? The main purpose of implementing the RecognitionListener interface in the mentioned application is speech recognition without displaying the Android / Google speech recognition dialog. Let's start with the onCreate method, here's SpeechRecognizer.createSpeechRecognizer (this);

used to initialize the SpeechRecognizer object. This SpeechRecognizer class is the one through which the speech recognition service can be accessed. This object is the main object in this implementation through which we start and stop the voice recognition process.
Another important thing is the display of speech visualization. Generally, when working with the default Google voice search dialog, while recording voice, the visualization is displayed around the microphone button. But now that we are conducting our own activity where the voice will be "captured", we must also apply voice visualization. This is achieved by the onRmsChanged (float rmsdB) method. An interesting observation about this method is that rmsdB is always in the range of -2 ~ to 10.0, it is not mentioned anywhere in the official documentation, it is just my personal observation and thus I set the maximum value of the progress bar in the onBeginningOfSpeech () method at 10. In the standard RecognitionListener stream whenever voice fluctuations are noticed, the rmsdB changes and the onRmsChanged method is called. In this implementation, I use this method to update the progress bar.
Next when the toggle button changes, example: SpeechRecognizer.startListening (recognizerIntent); and SpeechRecognizer.stopListening (); methods are called accordingly, with the appropriate RecognizerIntent.
Finally, one of the most important methods for designing an e-Dictate application is the onResults (Bundle results) method. Here, in this method, the speech recognition result is passed as an argument in the form of an ArrayList. After this, the result can be processed, as needed. In this case, the result is just displayed in TextView.
Layers of convolutional neural networks (CNN) were added before sending the characteristics to the transformer input. This makes it possible to reduce the difference in the dimensions of the input and output sequences (since the number of frames in the sound is significantly higher than the number of tokens in the text), which has a beneficial effect on training. This paper confirmed that transformers can indeed be used successfully in speech recognition!
To view the final version of the application and use it, visit the link:
https://play.google.com/store/apps/details?id=rs.edukom.diktat

5. CONCLUSION

This paper provides an overview of the achievements of machine learning in speech recognition and neural networks and (deep) generative modeling on a concrete example, which is a successful case of deep learning on an industrial scale. The paper analyzes the roles of generative models, emphasizing that the key advantages of incorporating knowledge of speech dynamics that are naturally enabled by deep generative modeling have yet to be incorporated as part of the new generation machine learning framework.
For speech recognition, one remaining future challenge lies in how to effectively integrate major relevant speech knowledge and problem constraints into new deep models of the future of JAVA programming. Examples of such knowledge and limitations would include distributed phonological representations of sound patterns of characteristics based on characteristics through a hierarchical structure based on modern phonology, articulatory dynamics and control of motor program, mechanisms of acoustic distortion to create noisy speech in speaker environments, lombardian effects noise-induced communication efficiency, and so on. Deep generative models are much more capable of imposing the above constraints than purely discriminatory DNNs. These deep generative models need to be parameterized to allow for extremely regular, matrix-oriented computing, modern high-efficiency programming that has already proven to be extremely fruitful for DNN. The design of the overall deep computational network architecture of the future can be motivated by approximate inference algorithms associated with the initial generative model. Discriminative learning algorithms such as backpropagation can then be developed and applied to learn all network parameters (i.e., large matrices) from end to end. Finally, the calculation of the execution time follows the inference algorithm in the generative model, but the parameters are taught to best discriminate all JAVA classes of speech sounds. This is similar to discriminatory learning for GMM-HMM,
Solving NLP problems using an embedded machine learning scheme can become promising when problems are part of broader big data analytics applications, where there can be not only words and other language entities but also business activities, people, events, and so on. embedded in a single vector space.

LITERATURE

Dudley, H. (1939).“The vocoder,” Bell Labs Rec., Vol. 17, pp. 122 6,
Dudley, H., Ryes, R. R., & Watkins, S. A. (1939). A synthetic speaker,” J. Franklin Inst., Vol. 227, pp. 739 64,
Davis, K. H., Biddulph, R., & Balashek, S. (1952). Automatic recognition of spoken digits,” J Acoust Soc Amer., vol. 24, no. 6, pp. 637 42, Nov.
Olson, H. F., & Belar, H. (1956). “Phonetic typewriter,” J Acoust Soc Amer., vol. 28, no. 6, pp. 1072 81, Nov.
Fry, D. B.(1959). “Theoretical aspects of the mechanical speech recognition,” J. Br. Inst. Radio Eng., Vol. 19, no. 4, pp. 211 29
Vintsyuk, T. K. (1968). “Speech discrimination by dynamic programming,” Kibernetika, Vol. 4, pp. 81 8,

Rabiner, L. R., S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, (1979). "Speaker independent recognition of isolated words using clustering techniques," IEEE Trans. Acoust. Speech Signal Process., Vol. 27, pp. 336 49

Wilpon, J. G., L. R. Rabiner, C. H. Lee, and E. Goldman, (1990). "Automatic recognition of keywords in unconstrained speech using hidden Markov models," IEEE Trans. Acoust. Speech Signal Process., Vol. 38, pp. 1870

Sahoo, S. K., T. Choubisa, and S. R. M. Prasanna, (2012). "Multimodal biometric person authentication: a review," IETE Tech. Rev., Vol. 29, no. 1, pp. 54 75

Pati, D., and S. R. M. Prasanna, (2010). "Speaker recognition from excitation source perspective," IETE Tech. Rev., Vol. 27, no. 2, pp. 138 57

Jayanna, H. S., and S. R. M. Prasanna, (2009). Analysis, feature extraction, modeling and testing techniques for speaker recognition," IETE Tech. Rev., Vol. 26, no. 3, pp. 181 90, Sep.

Sachin Singh, Manoj Tripathy, and R. S. Anand, (2014). "Subjective and objective analysis of speech enhancement algorithms for single channel speech patterns of Indian and English languages," IETE Tech. Rev., Vol. 31, no. 1, pp. 34 46,

Tom M. Mitchell, (1997). Machine Learning, New York, NY: McGraw Hill, International Edition

Baker, J. (1976). "Stochastic modeling for automatic speech recognition," in Speech Recognition. R. Reddy, Ed. New York, NY: Academic Press, pp. 297 307.

Jelinek, F. (1976). "Continuous speech recognition by statistical methods," Proc. IEEE, Vol. 64, no. 4, pp. 532 57,

Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, and D. OShgughnessy, (2009). "Research developments and directions in speech recognition and understanding. Part I," IEEE Signal Process. Mag., Vol. 26, no. 3, pp. 75 80,

Rabiner, L., and B.-H. Juang, (1993). Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliffs, NJ,

Juang, B.-H., S. E. Levinson, and M. M. Sondhi, (1986). "Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains," IEEE Trans. Inf. Theory, Vol. 32, no. 2, pp. 307 9

Deng, L., P. Kenny, M. Lennig, V. Gupta, F. Seitz, and P. Mermelsten, (1991). "Phonemic hidden Markov models with continuous mixture output densities for large vocabulary word recognition," IEEE Trans. Acoust. Speech Signal Process., Vol. 39, no. 7, pp. 1677 81

Bilmes, (2006)."What HMMs can do," IEICE Trans. Inf. Syst., Vol. E89-D, no. 3, pp. 869 91, Mar.

Bourlard, H., and C. J. Wellekens, (1989). "Links between Markov models and multilayer perceptrons," in Advances in Neural Information Processing, D.S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, pp. 502 10.

Morgan, N. and H. Bourlard, (1990). "Continuous speech recognition using multilayer perceptrons with hidden Markov models," in Proceedings of the IEEE International Conference ASSP, Albuquerque, NM, pp. 413 6.

Morgan, H. Hermansky, H. Bourlard, P. Kohn, and C. Wooters, (1991). "Continuous speech recognition using PLP analysis with multilayer perceptrons," in Proceedings of the IEEE International Conference ASSP, Toronto, ON, pp. 49 52.

Stadermann, J. and G. Rigoll, (2004). "A hybrid SVM/HMM acoustic modeling approach to automatic speech recognition," in Proceedings of the Interspeech, Jeju island, Korea, pp. 661 4.

Zhang, S. A. Ragni, and M. Gales, (2010). "Structured log linear models for noise robust speech recognition," IEEE Signal Process. Lett., Vol. 17, pp. 945 8, Nov. 2010.

Landauer, T. K., C. A. Kamm, and S. Singhal, (1987). "Teaching a minimally structured back propagation network to recognize speech," in Proceedings of the Ninth Annual Conference of the Cognitive Science Society, Seattle,Washington, pp. 531 6.

Transformer-based Acoustic Modeling for Hybrid Speech Recognition (https://arxiv.org/abs/1910.09799).