ISSN:0975-9646

Speech Recognition using Deep Learning

Akhilesh Halageri , Amrita Bidappa , Arjun C ,. Madan Mukund Sarathy ,. Shabana Sultana

Department of Computer Science and Engineering
The National Institute of Engineering,
Mysore,Karnataka, India

***Abstract*— Speech Recognition is the translation of spoken words into text. Speech recognition involves capturing and digitizing the sound waves, converting them to basic language units or phonemes, constructing words from phonemes, and contextually analyzing the words to ensure correct spelling for words that sound alike. The main purpose of the paper is to review the pattern matching abilities of neural networks on speech signal.**

***Keywords*— Speech Recognition, Neural Networks, Deep Learning, Machine Learning, Speech-to-text.**

I. INTRODUCTION

Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format. Many speech recognition applications, such as voice dialing, simple data entry and speech-to-text are in existence today.

Automatic speech recognition systems involve numerous separate components drawn from many different disciplines such as statistical pattern recognition, communication theory, signal processing, combinatorial mathematics, and linguistics. Speech recognition is an alternative to traditional methods of interacting with a computer, such as textual input through a keyboard. An effective system can replace, or reduce the reliability on, standard keyboard input

Attempts to build automatic speech recognition (ASR) systems were first made in the 1950s. These early speech recognition systems tried to apply a set of grammatical and syntactical rules to identify speech. If the spoken words adhered to a certain rule set, the system could recognize the words. However, human language has numerous exceptions to its own rules. The way words and phrases are spoken can be vastly altered by accents, dialects and mannerisms. Therefore, to achieve ASR we make use of Deep Learning Algorithm.

II. REVIEW

A. Existing Method:

The existing systems for ASR use complex statistical models. Hidden Markov Models have been very successful. These are statistical models that output a sequence of symbols or quantities. GMM-HMMs are used in speech recognition because a speech signal can be viewed as a piecewise stationary signal or a short-time stationary signal. Another reason why GMM-HMMs are popular is because they can be trained automatically and are simple and computationally feasible to use. But GMM-HMMs make various assumptions about the speech and as a result fail to generalize.

The disadvantages are:

- It is expensive, both in terms of memory and compute time.
- GMMs are statistically inefficient for modelling data that lie on or near a nonlinear manifold in the data space [1].
- The HMM needs to be trained on a set of seed sequences and generally requires a larger seed.
- For a given set of seed sequences, there are many possible HMMs, and choosing one can be difficult

B. Proposed Method:

The proposed system is to use "learning" algorithms which aim to learn the features, without any assumptions. Recently, algorithms using neural networks have been very successful in pattern recognition tasks - largely owing to the increased computational power. In contrast to GMM-HMMs, neural networks make no assumptions about feature statistical properties and have several qualities making them attractive recognition models for speech recognition. When used to estimate the probabilities of a speech feature segment, neural networks allow discriminative training in a natural and efficient manner. Few assumptions on the statistics of input features are made with neural networks.

Advantages:

- Powerful.
- Self-adjusting.
- Sophisticated pattern recognition.

C. Feasibility:

The only significant requirement of ASR systems is the training data. The ongoing research in the field of pattern recognition has made this data available in large amounts, including voice data required by ASR systems.

Motivation to use deep learning in speech recognition:

- Can model high-dimensional, highly correlated features efficiently.
- Layered architecture with non-linear operations offers feature extraction to be integrated with acoustic modeling.
- Better representation ability with fewer parameters.

III. DESIGN & IMPLEMENTATION

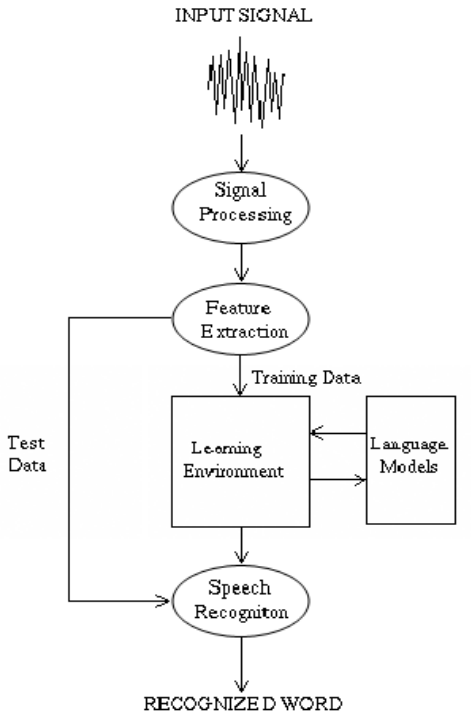

Fig.3.1 Architecture Diagram

A. *Preprocessing:*

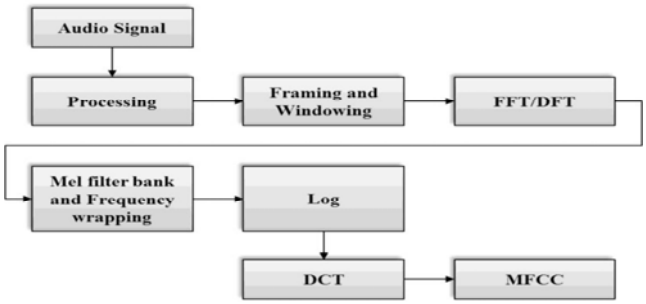

Fig.3.1.1 Feature Extraction

The main point to understand about speech is that the sounds generated by a human are filtered by the shape of the vocal tract including tongue, teeth etc. This shape determines what sound comes out. If we can determine the shape accurately, this should give us an accurate representation of the phoneme being produced. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the job of MFCCs is to accurately represent this envelope.

The features are extracted as follows:

- Frame the signal into short frames.
- Apply hamming window to make the signal periodic[2].
- Calculate the periodogram estimate of the power spectrum.
- Apply the mel filterbank to the power spectra, sum the energy in each filter.
- Take the logarithm of all filterbank energies.
- Take the DCT of the log filterbank energies.
- Keep DCT coefficients 2-13, discard the rest.
- Create a context window of adjacent frames to capture the phoneme context, further fed to neural network.

B. *Neural Networks*

The basic attributes of a neural network are

- A set of processing units.
- A set of connections.
- A computing procedure.
- A training procedure.

1) Processing Units: A neural network contains a potentially huge number of very simple processing units, roughly analogous to neurons in the brain. All these units operate simultaneously, supporting massive parallelism. All computation in the system is performed by these units; there is no other processor that oversees or coordinates their activity. At each moment in time, each unit simply computes a scalar function of its local inputs, and broadcasts the result (called the activation value) to its neighboring units. The units in a network are typically divided into input units, which receive data from the environment (such as raw sensory information); hidden units, which may internally transform the data representation; and/or output units, which represent decisions or control signals. The sample *from 0*-25ms is taken, 3 samples before that are added and three samples after that are added to generate a 1x91 mfcc matrix. This step is followed for all the sample frames. And, the net samples are taken in steps of 10ms. ie. 0-25ms, 10-35ms, 20-45ms and so on. This along with the respective audio files and the respective phoneme transcriptions are fed into the neural network.

2) Connections: The units in a network are organized into a given topology by a set of connections, or weights, shown as lines in a diagram. Each weight has a real value, typically ranging from to $-\infty$ to $+\infty$, although sometimes the range is limited. The value (or strength) of a weight describes how much influence a unit has on its neighbor; a positive weight causes one unit to excite another, while a negative weight causes one unit to inhibit another. Weights are usually one-directional (from input units towards output units), but they may be two-directional (especially when there is no distinction between input and output units). The values of all the weights predetermine the network's computational reaction to any arbitrary input pattern; thus the weights encode the long-term memory, or the knowledge, of the network. Weights can change as a result of training, but they tend to change slowly, because accumulated knowledge changes slowly. This is in contrast to activation patterns, which are transient functions of the current input, and so are a kind of short-term memory.

Here, three hidden layers with 100 units each with 91input units and 43 output units are used.

3) Computation: Computation always begins by presenting an input pattern to the network, or clamping a pattern of activation on the input units. Then the activations of all of the remaining units are computed, either synchronously (all at once in a parallel system) or asynchronously (one at a time, in either randomized or natural order), as the case may be. In unstructured networks, this process is called spreading activation; in layered networks, it is called forward propagation, as it progresses from the input layer to the output layer. In feedforward networks (i.e., networks without feedback), the activations will stabilize as soon as the computations reach the output layer; but in recurrent networks (i.e., networks with feedback), the activations may never stabilize, but may instead follow a dynamic trajectory through state space, as units are continuously updated.

4) Training: Training a network, in the most general sense, means adapting its connections so that the network exhibits the desired computational behavior for all input patterns. The process usually involves modifying the weights (moving the hyperplanes/hyperspheres); but sometimes it also involves modifying the actual topology of the network, i.e., adding or deleting connections from the network (adding or deleting hyperplanes/hyperspheres). In a sense, weight modification is more general than topology modification, since a network with abundant connections can learn to set any of its weights to zero, which has the same effect as deleting such weights. However, topological changes can improve both generalization and the speed of learning, by constraining the class of functions that the network is capable of learning [3]. This can be controlled by adjusting learning rate and momentum.

*5) Training procedure:*Perceptrons are the simplest type of feedforward networks that use supervised learning. A perceptron is comprised of binary threshold units arranged into layers. Multi-layer perceptrons (MLPs) can theoretically learn any function, but they are more complex to train. The Delta Rule cannot be applied directly to MLPs because there are no targets in the hidden layer(s). However, if an MLP uses continuous rather than discrete activation functions (i.e., sigmoids rather than threshold functions), then it becomes possible to use partial derivatives and the chain rule to derive the influence of any weight on any output activation, which in turn indicates how to modify that weight in order to reduce the network's error. This generalization of the Delta Rule is known as backpropagation.
Backpropagation, an abbreviation for "backward propagation of errors", is a common method of training artificial neural networks used in conjunction with an optimization method such as gradient descent. The method calculates the gradient of a loss function with respects to all the weights in the network. The gradient is fed to the optimization method which in turn uses it to update the weights, in an attempt to minimize the loss function. Backpropagation requires that the activation function used by the artificial neurons (or "nodes") be differentiable.

6) Algorithm for a 3-layer network (only one hidden layer) [4]:

initialize network weights (often small random values)
 do:
  forEach training example ex
   prediction = neural-net-output(network, ex)
   actual = teacher-output(ex)
   compute error (prediction - actual) at the output units
   compute $\Delta\omega_i$ for all weights from hidden layer to output layer
   compute $\Delta\omega_i$ for all weights from input layer to hidden layer
   update network weights
 until all examples classified correctly or another stopping criterion satisfied
 return the network

After training the neural network, the input can be given to the network which gives the phoneme sequence as the output.

C. Word extraction from Phoneme Sequence:

The phoneme activations are fed to the word and syntax recognition part of the recognition system, which is based on a dynamic programming (DP) procedure to find the best path through a phoneme network. The network defines possible word sequences at the phoneme level. Optional pronunciations are realized as parallel branches. Inhalation sounds before the utterance and short silent intervals at word boundaries are included as optional branches in the net. Phoneme duration information is used explicitly in the DP-algorithm to limit the search. Within the duration limits, uniform distribution densities are assumed. These limits are quite wide, and therefore probably don't influence the recognition result in a significant way. However, the algorithm is designed for more extensive use of duration information in the future.
Simple neural networks trained on a small speech corpus of isolated words outperformed GMM-HMM models, efficiently mapping single isolated words to relevant text. They can classify a dictionary of words directly without the intermediate phoneme representation. But for continuous speech recognition the neural network architecture would be complex (time-delay neural nets or recurrent neural nets) and the data required would be in the orders of gigabytes if not terabytes.

ADVANTAGES

- Neural networks can be taught to map an input space to any kind of output space. They are simple and intuitive, hence they are commonly used.
- They are naturally discriminative.
- They are modular in design, so they can be easily combined into larger systems.
- They have a probabilistic interpretation, so they can be easily integrated with statistical techniques like HMMs

IV. CONCLUSION

One may well ask whether adequate ASR will ever truly be accomplished. In general, one may assume that almost all artificial intelligence (AI) tasks are potentially feasible; certainly great progress in chess-playing machines and robotics supports this view. Compare ASR to the task of automatically driving a car; the latter requires intelligent interpretation of the field of vision for cameras mounted on a vehicle. While algorithms needed for cars would be very different than for ASR, there are similarities in signal processing and both challenges seem daunting (i.e., replacing a human driver with a similar-performing algorithm might seem as far-fetched as having a fully understanding ASR device). It would seem that ASR is much closer to potential solution, however.

Future Enhancements

- SUI – Speech-based User Interfaces can be developed.
- Greater accuracy in recognising words can be obtained.
- Greater system control/commands can be included.
- More compatible software

REFERENCES

[1] Speech Recognition - A Deep Learning Approach, Dong Yu, Li Deng, Microsoft Research, ISBN 978-1-4471-5779-3
[2] Think DSP Digital Signal Processing in Python, Allen B. Downey
[3] Speech Recognition using Neural Networks, Joe Tebelskis, May 1995, CMU-CS-95-142
[4] Wikipedia - en.wikipedia.org/wiki/Backpropagation