Speech Recognition System

Chapter · July 2012

CITATIONS
0

READS
4,138

1 author:

Ahmed Moawad
Information Technology Institute
1 PUBLICATION **0** CITATIONS

SEE PROFILE

chapter 2

Speech Recognition

2.1 | INTRODUCTION

Biometrics is, in the simplest definition, something you are. It is a physical characteristic unique to each individual such as fingerprint, retina, iris, speech. Biometrics has a very useful application in security; it can be used to authenticate a person's identity and control access to a restricted area, based on the premise that the set of these physical characteristics can be used to uniquely identify individuals. Speech signal conveys two important types of information, the primarily the speech content and on the secondary level, the speaker identity. Speech recognizers aim to extract the lexical information from the speech signal independently of the speaker by reducing the inter-speaker variability. On the other hand, speaker recognition is concerned with extracting the identity of the person speaking the utterance. So both speech recognition and speaker recognition system is possible from same voice input.

We use in our project the speech recognition technique because we want in our project to recognize the word that the stick will make action depending on this word.

Mel Filter Cepstral Coefficient (MFCC) is used as feature for both speech and speaker recognition. We also combined energy features and delta and delta-delta features of energy and MFCC. After calculating feature, neural networks are used to model the speech recognition. Based on the speech model the system decides whether or not the uttered speech matches what was prompted to utter.

2.2 | LITERATURE REVIEW

2.2.1 | Pattern Recognition

Pattern recognition, one of the branches of artificial intelligence, sub-section of machine learning, is the study of how machines can observe the environment, learn to distinguish patterns of interest from their background, and make sound and reasonable decisions about the categories of the patterns. A pattern can be a fingerprint image, a handwritten cursive word, a human face, or a speech signal, sales pattern etc…

The applications of pattern recognition include data mining, document classification, financial forecasting, organization and retrieval of multimedia databases, and biometrics (personal identification based on various physical attributes such as face, retina, speech, ear and fingerprints).The essential steps of

pattern recognition are: Data Acquisition, Preprocessing, Feature Extraction, Training and Classification.

Features are used to denote the descriptor. Features must be selected so that they are discriminative and invariant. They can be represented as a vector, matrix, tree, graph, or string.

They are ideally similar for objects in the same class and very different for objects indifferent class. Pattern class is a family of patterns that share some common properties. Pattern recognition by machine involves techniques for assigning patterns to their respective classes automatically and with as little human intervention as possible.

Learning and Classification usually use one of the following approaches: Statistical Pattern Recognition is based on statistical characterizations of patterns, assuming that the patterns are generated by a probabilistic system. Syntactical (or Structural) Pattern Recognition is based on the structural interrelationships of features. Given a pattern, its recognition/classification may consist of one of the following two tasks according to the type of learning procedure:

1) Supervised Classification (e.g., Discriminant Analysis) in which the input pattern is identified as a member of a predefined class.

2) Unsupervised Classification (e.g., clustering) in which the pattern is assigned to a previously unknown class.

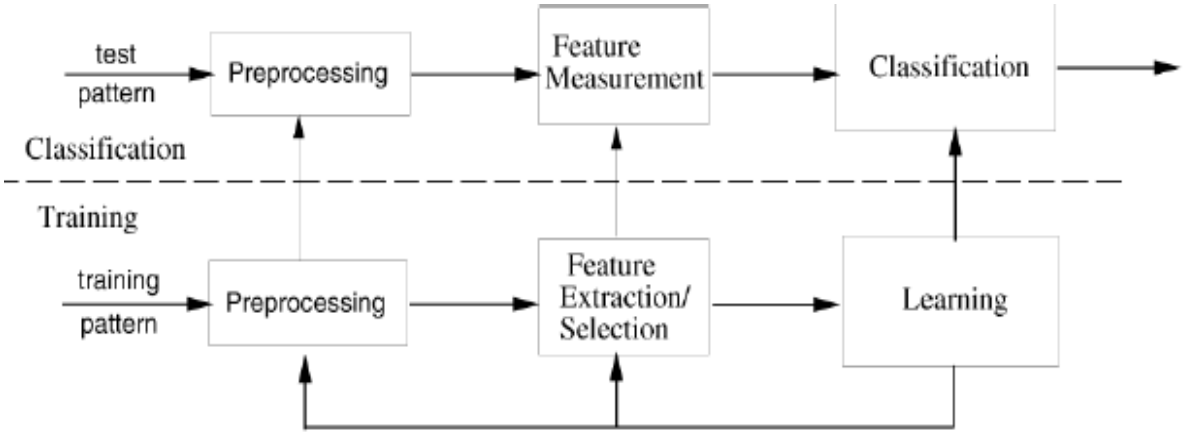

Fig. (2.1): General block diagram of pattern recognition system

2.2.2 | Generation of Voice

Speech begins with the generation of an airstream, usually by the lungs and diaphragm -process called initiation. This air then passes through the larynx tube, where it is modulated by the glottis (vocal chords). This step is called phonation or voicing, and is responsible fourth generation of pitch and tone. Finally, the modulated air is filtered by the mouth, nose, and throat - a process called articulation - and the resultant pressure wave excites the air.

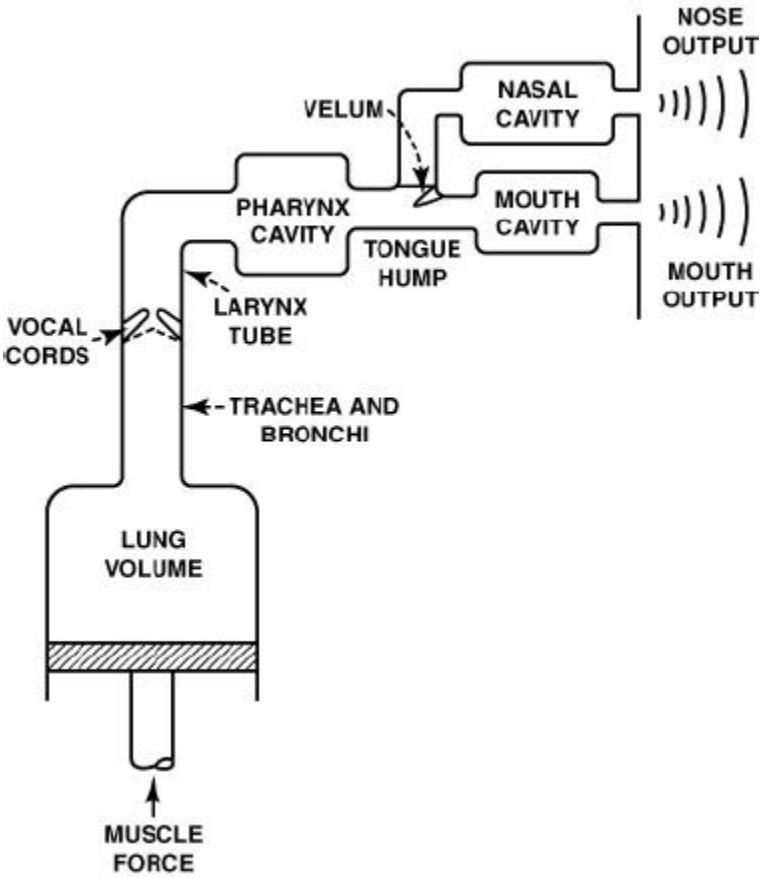

Fig. (2.2): Vocal Schematic

Depending upon the positions of the various articulators different sounds are produced. Position of articulators can be modeled by linear time- invariant system that has frequency response characterized by several peaks called formants. The change in frequency of formants characterizes the phoneme being articulated.

As a consequence of this physiology, we can notice several characteristics of the frequency domain spectrum of speech. First of all, the oscillation of the glottis

results in an underlying fundamental frequency and a series of harmonics at multiples of this fundamental. This is shown in the figure below, where we have plotted a brief audio waveform for the phoneme /i: / and its magnitude spectrum. The fundamental frequency (180 Hz) and its harmonics appear as spikes in the spectrum. The location of the fundamental frequency is speaker dependent, and is a function of the dimensions and tension of the vocal chords. For adults it usually falls between 100 Hz and 250 Hz, and females‟ average significantly higher than that of males.

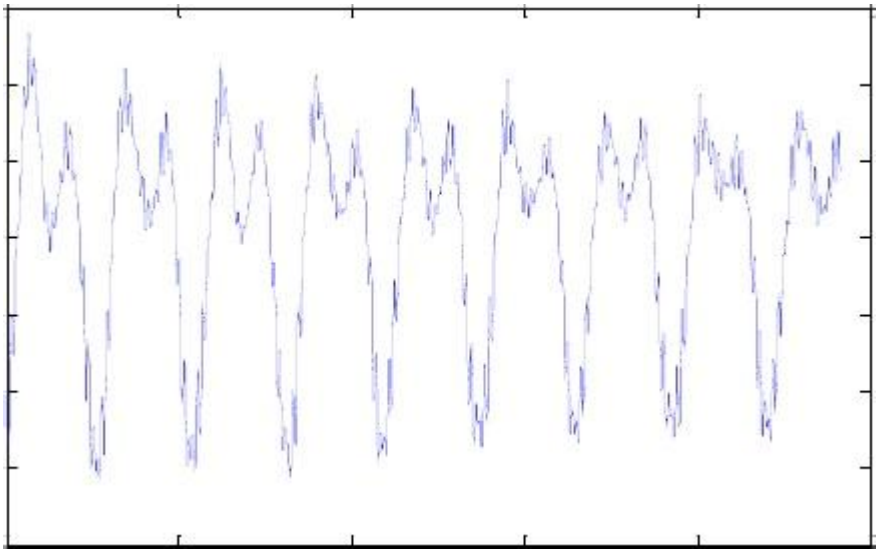

Fig. (2.3): Audio Sample for /i: / phoneme showing stationary property of phonemes for a short period

The sound comes out in phonemes which are the building blocks of speech. Each phoneme resonates at a fundamental frequency and harmonics of it and thus has high energy at those frequencies in other words have different formats. It is the feature that enables the identification of each phoneme at the recognition stage. The variations in

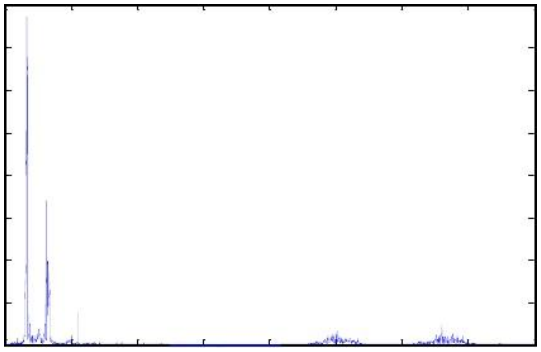

Fig.(2.4): Audio Magnitude Spectrum for /i:/ phoneme showing fundamental frequency and its harmonics

Inter-speaker features of speech signal during utterance of a word are modeled in word training in speech recognition. And for speaker recognition the intra-speaker variations in features in long speech content is modeled.

Besides the configuration of articulators, the acoustic manifestation of a phoneme is affected by:

- Physiology and emotional state of speaker.
- Phonetic context.
- Accent.

2.2.3 | Voice as Biometric

The underlying premise for voice authentication is that each person's voice differs in pitch, tone, and volume enough to make it uniquely distinguishable. Several factors contribute to this uniqueness: size and shape of the mouth, throat, nose, and teeth (articulators) and the size, shape, and tension of the vocal cords. The chance that all of these are exactly the same in any two people is very low. Voice Biometric has following advantages from other form of biometrics:

- Natural signal to produce
- Implementation cost is low since, doesn't require specialized input device
- Acceptable by user

Easily mixed with other form of authentication system for multifactor authentication only biometric that allows users to authenticate remotely.

2.2.4 | Speech Recognition

Speech is the dominant means for communication between humans, and promises to be important for communication between humans and machines, if it can just be made a little more reliable.

Speech recognition is the process of converting an acoustic signal to a set of words. The applications include voice commands and control, data entry, voice user interface, automating the telephone operator's job in telephony, etc. They can also serve as the input to natural language processing. There is two variant of speech recognition based on the duration of speech signal:

Isolated word recognition, in which each word is surrounded by some sort of pause, is much easier than recognizing continuous speech, in which words run into each other and have to be segmented. Speech recognition is a difficult task because

of the many source of variability associated with the signal such as the acoustic realizations of phonemes, the smallest sound units of which words are composed, are highly dependent on the context. Acoustic variability can result from changes in the environment as well as in the position and characteristics of the transducer. Third, within speaker variability can result from changes in the speaker's physical and emotional state, speaking rate, or voice quality. Finally, differences in socio linguistic background, dialect, and vocal tract size and shape can contribute to cross-speaker variability. Such variability is modeled in various ways. At the level of signal representation, the representation that emphasizes the speaker independent features is developed.

2.2.5 | Speaker Recognition

Speaker recognition is the process of automatically recognizing who is speaking on the basis of individual's information included in speech waves. Speaker recognition can be classified into identification and verification. Speaker recognition has been applied most often as means of biometric authentication.

2.2.5.1 | Types of Speaker Recognition

Speaker Identification

Speaker identification is the process of determining which registered speaker provides a given utterance. In Speaker Identification (SID) system, no identity claim is provided, the test utterance is scored against a set of known (registered) references for each potential speaker and the one whose model best matches the test utterance is selected. There is two types of speaker identification task closed-set and open-set speaker identification .In closed-set, the test utterance belongs to one of the registered speakers.

During testing, a matching score is estimated for each registered speaker. The speaker corresponding to the model with the best matching score is selected. This requires N comparisons for a population of N speakers. In open-set, any speaker can access the system; those who are not registered should be rejected. This requires another model referred to as garbage model or imposter model or background model, which is trained with data provided by other speakers different from the registered speakers.

During testing, the matching score corresponding to the best speaker model is compared with the matching score estimated using the garbage model. In order to accept or reject the speaker, making the total number of comparisons equal to N +

1. Speaker identification performance tends to decrease as the population size increases.

Speaker verification

Speaker verification, on the other hand, is the process of accepting or rejecting the identity claim of a speaker. That is, the goal is to automatically accept or reject an identity that is claimed by the speaker. During testing, a verification score is estimated using the claimed speaker model and the anti-speaker model. This verification score is then compared to a threshold. If the score is higher than the threshold, the speaker is accepted, otherwise, the speaker is rejected.

Thus, speaker verification, involves a hypothesis test requiring a simple binary decision: accept or reject the claimed identity regardless of the population size. Hence, the performance is quite independent of the population size, but it depends on the number of test utterances used to evaluate the performance of the system.

2.2.6 | Speaker/Speech Modeling

There are various pattern modeling/matching techniques. They include Dynamic Time Warping (DTW), Gaussian Mixture Model (GMM), Hidden Markov Modeling (HMM), Artificial Neural Network (ANN), and Vector Quantization (VQ). These are interchangeably used for speech, speaker modeling. The best approach is statistical learning methods: GMM for Speaker Recognition, which models the variations in features of a speaker for a long sequence of utterance.

And another statistical method widely used for speech recognition is HMM. HMM models the Markovian nature of speech signal where each phoneme represents a state and sequence of such phonemes represents a word. Sequence of Features of such phonemes from different speakers is modeled by HMM.

2.3 | IMPLEMENTATION DETAILS

The implementation of system includes common pre-processing and feature extraction module, speaker independent speech modeling and classification by ANNs.

2.3.1 | Pre-Processing and Feature Extraction

Starting from the capturing of audio signal, feature extraction consists of the following steps as shown in the block diagram below:

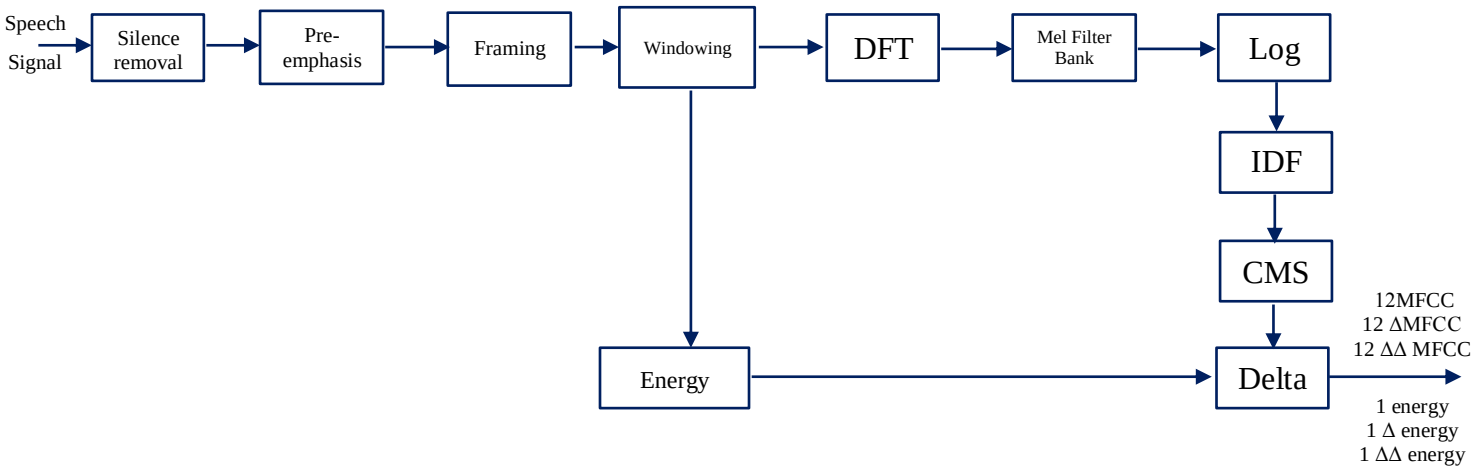

Fig. (2.5): Pre-Processing and Feature Extraction

2.3.1.1 | Capture

The first step in processing speech is to convert the analog representation (first air pressure, and then analog electric signals in a microphone) into a digital signal x[n], where n is an index over time. Analysis of the audio spectrum shows that nearly all energy resides in the band between DC and 4 kHz, and beyond 10 kHz there is virtually no energy what so ever.

Used sound format:

- 22050 Hz
- 16-bits, Signed
- Little Endian
- Mono Channel
- Uncompressed PCM

2.3.1.2 | End point detection and Silence removal

The captured audio signal may contain silence at different positions such as beginning of signal, in between the words of a sentence, end of signal…. etc. If silent frames are included, modeling resources are spent on parts of the signal which do not contribute to the identification. The silence present must be removed before further processing. There are several ways for doing this: most popular are Short Time Energy and Zeros Crossing Rate. But they have their own limitation regarding setting thresholds as an ad hocbasis. The algorithm we used uses

statistical properties of background noise as well as physiological aspect of speech production and does not assume any ad hoc threshold.

It assumes that background noise present in the utterances is Gaussian in nature. Usually first 200msec or more (we used 4410 samples for the sampling rate 22050samples/sec) of a speech recording corresponds to silence (or background noise) because the speaker takes some time to read when recording starts.

Endpoint Detection Algorithm:

Step 1:

Calculate the mean (μ) and standard deviation (σ) of the first 200ms samples of the given utterance. The background noise is characterized by this μ and σ.

Step 2:

Go from 1st sample to the last sample of the speech recording. In each sample, check whether one-dimensional Mahalanobis distance functions i.e. | x-μ |/ σ greater than 3 or not. If Mahalanobis distance function is greater than 3, the sample is to be treated as voiced sample otherwise it is an unvoiced/silence. The threshold reject the samples up to 99.7% as per given by P [|x−μ|≤3σ] =0.997 in a Gaussian distribution thus accepting only the voiced samples.

Step 3:

Mark the voiced sample as 1 and unvoiced sample as 0. Divide the whole speech signal into 10 ms non-overlapping windows. Represent the complete speech by only zeros and ones.

Step 4:

Consider there are M number of zeros and N number of ones in a window. If M ≥ N then convert each of ones to zeros and vice versa. This method adopted here keeping in mind that a speech production system consisting of vocal cord, tongue, vocal tract etc. cannot change abruptly in a short period of time window taken here as 10ms.

Step 5:

Collect the voiced part only according to the labeled „1“ samples from the windowed array and dump it in a new array. Retrieve the voiced part of the original speech signal from labeled 1 sample.

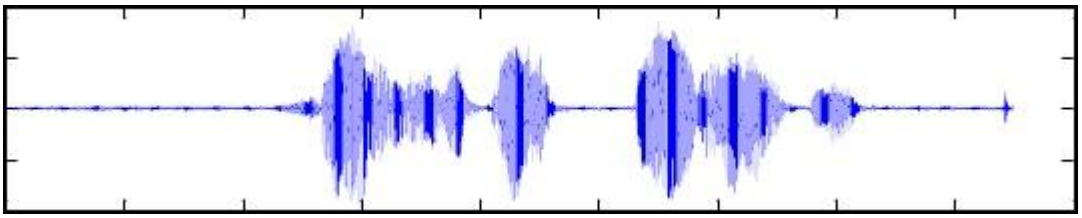

Fig. (2.6): Input signal to End-point detection system

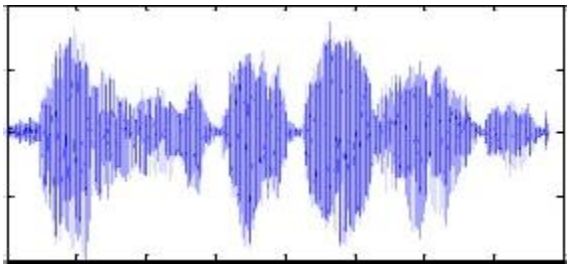

Fig. (2.7): Output signal from End point Detection System

2.3.1.3 | PCM Normalization

The extracted pulse code modulated values of amplitude is normalized, to avoid amplitude variation during capturing.

2.3.1.4 | Pre-emphasis

Usually speech signal is pre-emphasized before any further processing, if we look at the spectrum for voiced segments like vowels, there is more energy at lower frequencies than the higher frequencies. This drop in energy across frequencies is caused by the nature of the glottal pulse. Boosting the high frequency energy makes information from these higher formants more available to the acoustic model and improves phone detection accuracy. The pre-emphasis filter is a first-order high-pass filter. In the time domain, with input x[n]and $0.9 \leq \alpha \leq 1.0$, the filter equation is:

$$y[n] = x[n] - \alpha\, x[n-1]$$

We used $\alpha=0.95$.

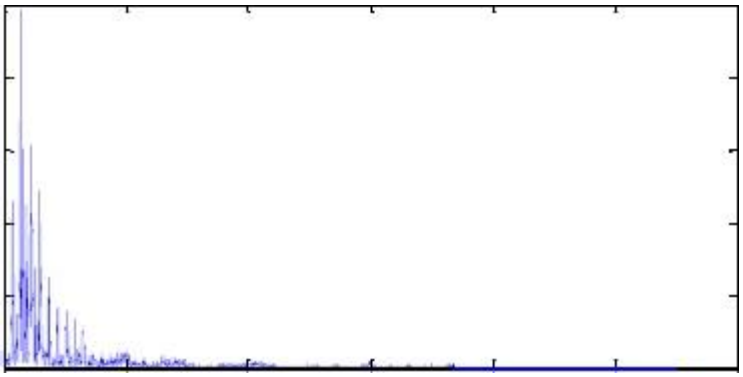

Fig. (2.8): Signal before Pre-Emphasis

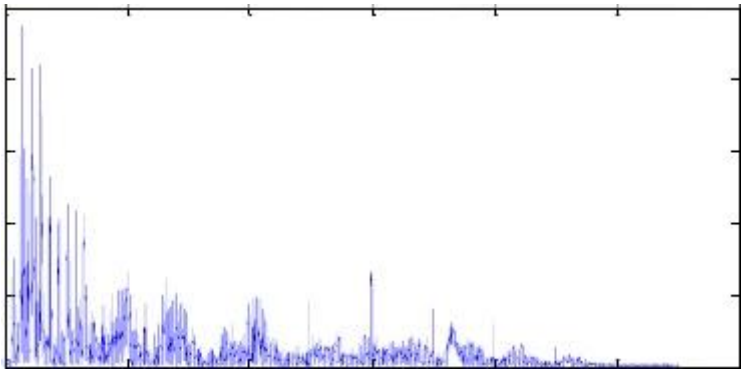

Fig.(2.9): Signal after Pre-Emphasis

2.3.1.5 | Framing and windowing

Speech is a non-stationary signal, meaning that its statistical properties are not constant across time. Instead, we want to extract spectral features from a small window of speech that characterizes a particular sub phone and for which we can make the (rough) assumption that the signal is stationary (i.e. its statistical properties are constant within this region).We used frame block of 23.22ms with 50% overlapping i.e., 512 samples per frame.

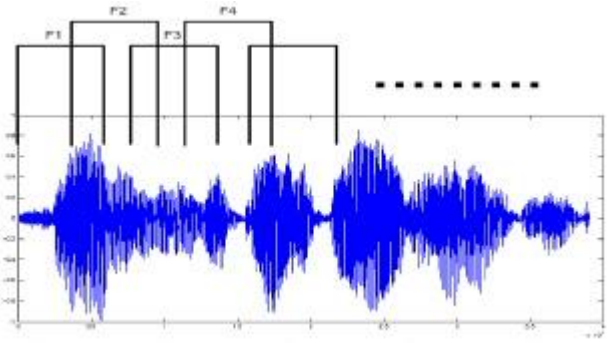

Fig.(2.10): Frame Blocking of the Signal

The rectangular window (i.e., no window) can cause problems, when we do Fourier analysis; it abruptly cuts of the signal at its boundaries. A good window function has a narrow main lobe and low side lobe levels in their transfer functions, which shrinks the values of the signal toward zero at the window boundaries, avoiding discontinuities. The most commonly used window function in speech processing is the Hamming window defined as follows:

$$S_w(n) = \{0.54 - 0.46 \cos\left(\frac{2\pi(n-1)}{N-1}\right)\}, 1 \leq n \leq N$$

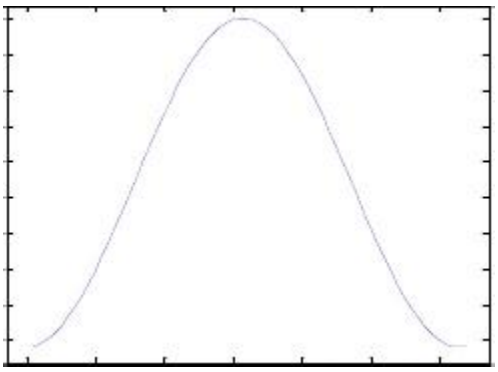

Fig.(2.11): Hamming window

The extraction of the signal takes place by multiplying the value of the signal at time n, s frame [n], with the value of the window at time n, S_w [n]:

$$Y[n] = S_w[n] \times S_{frame}[n]$$

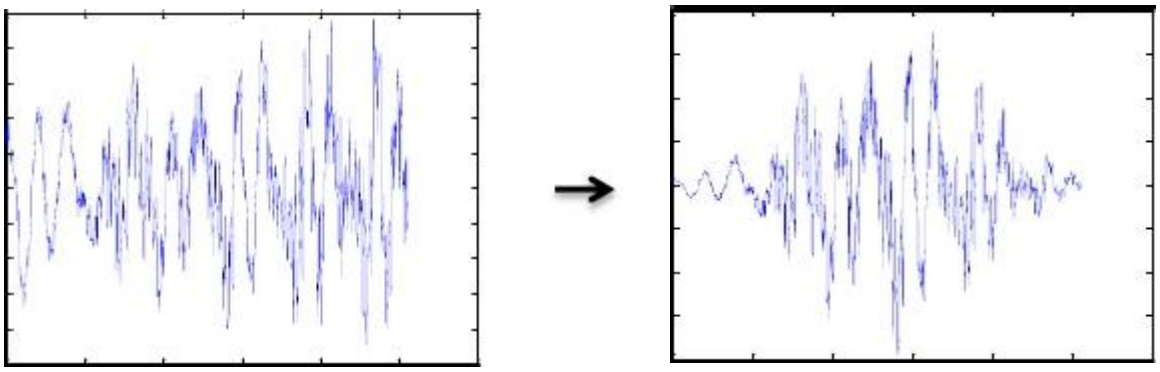

Fig.(2.12): A single frame before and after windowing

2.3.1.6 | Discrete Fourier Transform

A Discrete Fourier Transform (DFT) of the windowed signal is used to extract the frequency content (the spectrum) of the current frame. The tool for extracting spectral information i.e., how much energy the signal contains at discrete frequency bands for a discrete-time (sampled) signal is the Discrete Fourier Transform or DFT. The input to the DFT is a windowed signal x[n]...x[m], and the output, for each of N discrete frequency bands, is a complex number X[k] representing the magnitude and phase of that frequency component in the original signal.

$$bin_k = \left| \sum_{n=1}^{N} S_w(n) e^{-i(n-1)k\frac{2\pi}{N}} \right| , k = 0,1,2, \dots \dots , N-1$$

The commonly used algorithm for computing the DFT is the Fast Fourier Transform or in short FFT.

2.3.1.7 | Mel Filter

For calculating the MFCC, first, a transformation is applied according to the following formula:

$$Mel(x) = 2595 \log[1 + \frac{x}{700}]$$

Where, x is the linear frequency. Then, a filter bank is applied to the amplitude of the Mel-scaled spectrum. The Mel frequency warping is most conveniently done by utilizing a filter bank with filters centered according to Mel

frequencies. The width of the triangular filters varies according to the Mel scale, so that the log total energy in a critical band around the center frequency is included. The centers of the filters are uniformly spaced in the Mel scale.

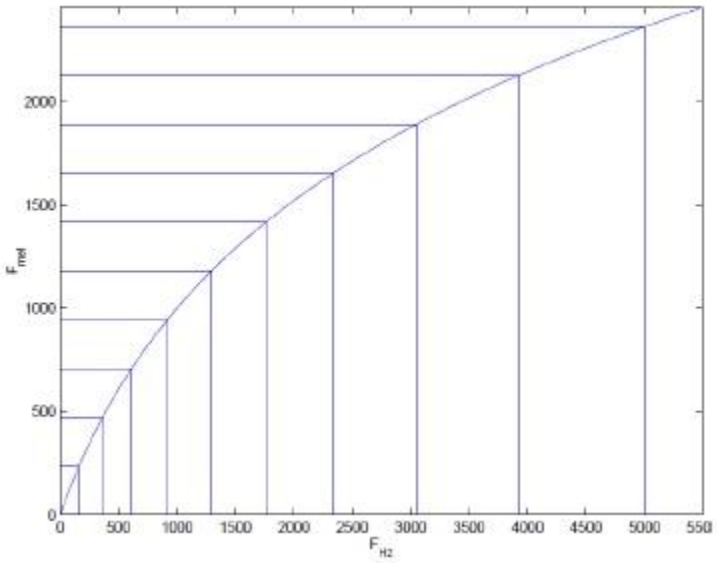

Fig.(2.13): Equally spaced Mel values

The result of Mel filter is information about distribution of energy at each Mel scale band. We obtain a vector of outputs (12 coeffs.) from each filter.

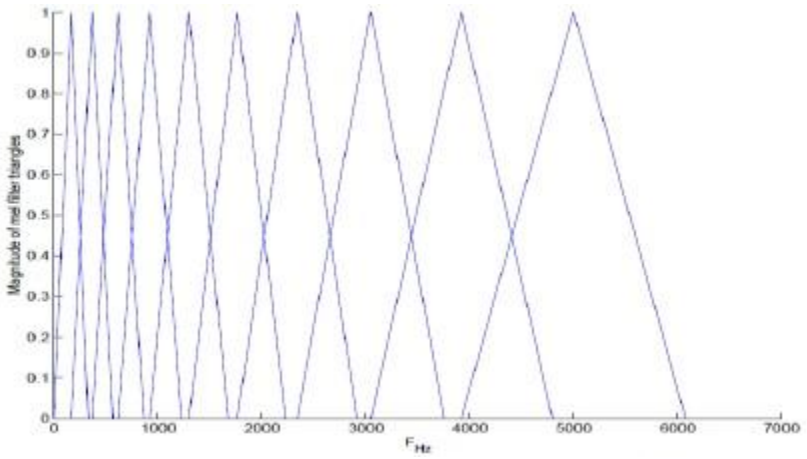

Fig.(2.13): Triangular filter bank in frequency scale

We have used 30 filters in the filter bank.

2.3.1.8 | Cestrum by Inverse Discrete Fourier Transform

Cestrum transform is applied to the filter outputs in order to obtain MFCC feature of each frame. The triangular filter outputs Y (i), i=0, 1, 2… M are compressed using logarithm, and discrete cosine transform (DCT) is applied. Here, M is equal to number of filters in filter bank i.e., 30.

$$c[n] = \sum_{i=1}^{M} \log Y(i) \cos\left[\frac{\pi n}{M}(i - \frac{1}{2})\right]$$

Where, C[n] is the MFCC vector for each frame.

The resulting vector is called the Mel-frequency cepstrum (MFC), and the individual components are the Mel-frequency Cepstral coefficients (MFCCs). We extracted 12 features from each speech frame.

2.3.1.9 | Post Processing

Cepstral Mean Subtraction (CMS)

A speech signal may be subjected to some channel noise when recorded, also referred to as the channel effect. A problem arises if the channel effect when recording training data for a given person is different from the channel effect in later recordings when the person uses the system. The problem is that a false distance between the training data and newly recorded data is introduced due to the different channel effects. The channel effect is eliminated by subtracting the Mel-cepstrum coefficients with the mean Mel-cepstrum coefficients:

$$mc_j(q) = C_i(q) - \frac{1}{M}\sum_{i=1}^{M} c_i(q), \;\; q = 1,2,3, \ldots\ldots,12$$

The energy feature

The energy in a frame is the sum over time of the power of the samples in the frame; thus for a signal x in a window from time sample t_1 to time sample t_2 the energy is:

$$Energy = \sum_{t=t_2}^{t_2} X^2[t]$$

Delta feature

Another interesting fact about the speech signal is that it is not constant from frame to frame. Co-articulation (influence of a speech sound during another

adjacent or nearby speech sound) can provide a useful cue for phone identity. It can be preserved by using delta features. Velocity (delta) and acceleration (delta delta) coefficients are usually obtained from the static window based information. This delta and delta delta coefficients model the speed and acceleration of the variation of Cepstral feature vectors across adjacent windows. A simple way to compute deltas would be just to compute the difference between frames; thus the delta value d(t) for a particular Cepstral value c (t) at time t can be estimated as:

$$d(t) = \Delta f_k[i] = f_{k+M}[i] - f_{k-M}[i]$$

The differentiating method is simple, but since it acts as a high-pass filtering operation on the parameter domain, it tends to amplify noise. The solution to this is linear regression, i.e. first-order polynomial, the least squares solution is easily shown to be of the following form:

$$\Delta f_k[i] = \frac{\sum_{m=-M}^{M} m f_{k+m}[i]}{\sum_{m=-M}^{M} m^2}$$

Where, M is regression window size. We used M=4.

Composition of Feature Vector

We calculated 39 Features from each frame:

- 12 MFCC Features.
- 12 Deltas MFCC.
- 12 Delta-Deltas MFCC.
- 1 Energy Feature.
- 1 Delta Energy Feature.
- 1 Delta-Delta Energy Feature.

2.4 | ARTIFICIAL NEURAL NETWORKS

2.4.1 | Introduction

We have used ANNs to model our system and train voices and test it to classify it into words categories which return actions. And here we will make an overview about **artificial neural networks**.

The original inspiration for the term Artificial Neural Network came from examination of central nervous systems and their neurons, axons, dendrites, and synapses, which constitute the processing elements of biological neural networks investigated by neuroscience. In an artificial neural network, simple artificial nodes, variously called "neurons", "neurodes", "processing elements" (PEs) or

"units", are connected together to form a network of nodes mimicking the biological neural networks — hence the term "artificial neural network".

Because neuroscience is still full of unanswered questions, and since there are many levels of abstraction and therefore many ways to take inspiration from the brain, there is no single formal definition of what an artificial neural network is. Generally, it involves a network of simple processing elements that exhibit complex global behavior determined by connections between processing elements and element parameters. While an artificial neural network does not have to be adaptive per se, its practical use comes with algorithms designed to alter the strength (weights) of the connections in the network to produce a desired signal flow.

These networks are also similar to the biological neural networks in the sense that functions are performed collectively and in parallel by the units, rather than there being a clear delineation of subtasks to which various units are assigned (see also connectionism). Currently, the term Artificial Neural Network (ANN) tends to refer mostly to neural network models employed in statistics, cognitive psychology and artificial intelligence. Neural network models designed with emulation of the central nervous system (CNS) in mind are a subject of theoretical neuroscience and computational neuroscience.

In modern software implementations of artificial neural networks, the approach inspired by biology has been largely abandoned for a more practical approach based on statistics and signal processing. In some of these systems, neural networks or parts of neural networks (such as artificial neurons) are used as components in larger systems that combine both adaptive and non-adaptive elements. While the more general approach of such adaptive systems is more suitable for real-world problem solving, it has far less to do with the traditional artificial intelligence connectionist models. What they do have in common, however, is the principle of non-linear, distributed, parallel and local processing and adaptation. Historically, the use of neural networks models marked a paradigm shift in the late eighties from high-level (symbolic) artificial intelligence, characterized by expert systems with knowledge embodied in if-then rules, to low-level (sub-symbolic) machine learning, characterized by knowledge embodied in the parameters of a dynamical system.

2.4.2 | Models

Neural network models in artificial intelligence are usually referred to as artificial neural networks (ANNs); these are essentially simple mathematical models defining a function or a distribution over or both and , but sometimes models are also intimately associated with a particular learning algorithm or learning rule. A common use of the phrase ANN model really means the definition of a class of such functions (where members of the class are obtained by varying parameters, connection weights, or specifics of the architecture such as the number of neurons or their connectivity).

2.4.3 | Network Function

The word network in the term 'artificial neural network' refers to the inter–connections between the neurons in the different layers of each system. An example system has three layers. The first layer has input neurons, which send data via synapses to the second layer of neurons, and then via more synapses to the third layer of output neurons. More complex systems will have more layers of neurons with some having increased layers of input neurons and output neurons. The synapses store parameters called "weights" that manipulate the data in the calculations. An ANN is typically defined by three types of parameters:

- The interconnection pattern between different layers of neurons
- The learning process for updating the weights of the interconnections
- The activation function that converts a neuron's weighted input to its output activation.

Mathematically, a neuron's network function is defined as a composition of other functions, which can further be defined as a composition of other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables. A widely used type of composition is the nonlinear weighted sum, where (commonly referred to as the activation function) is some predefined function, such as the hyperbolic tangent. It will be convenient for the following to refer to a collection of functions as simply a vector.

2.4.4 | ANN dependency graph

This figure depicts such a decomposition of , with dependencies between variables indicated by arrows. These can be interpreted in two ways.

The first view is the functional view: the input is transformed into a 3-dimensional vector , which is then transformed into a 2-dimensional vector , which is finally transformed into . This view is most commonly encountered in the context of optimization.

The second view is the probabilistic view: the random variable depends upon the random variable , which depends upon , which depends upon the random variable . This view is most commonly encountered in the context of graphical models.

The two views are largely equivalent. In either case, for this particular network architecture, the components of individual layers are independent of each other (e.g., the components of are independent of each other given their input). This naturally enables a degree of parallelism in the implementation. Two separate depictions of the recurrent ANN dependency graph.

Networks such as the previous one are commonly called feed forward, because their graph is a directed acyclic graph. Networks with cycles are commonly called recurrent. Such networks are commonly depicted in the manner shown at the top of the figure, where is shown as being dependent upon itself. However, an implied temporal dependence is not shown.

2.4.5 | Learning

What has attracted the most interest in neural networks is the possibility of learning. Given a specific task to solve, and a class of functions, learning means using a set of observations to find which solves the task in some optimal sense.

This entails defining a cost function such that, for the optimal solution, - i.e., no solution has a cost less than the cost of the optimal solution (see Mathematical optimization).

The cost function is an important concept in learning, as it is a measure of how far away a particular solution is from an optimal solution to the problem to be solved. Learning algorithms search through the solution space to find a function that has the smallest possible cost.

For applications where the solution is dependent on some data, the cost must necessarily be a function of the observations; otherwise we would not be modeling anything related to the data. It is frequently defined as a statistic to which only approximations can be made. As a simple example, consider the problem of finding the model , which minimizes , for data pairs drawn from some distribution . In practical situations we would only have samples from and thus, for the above example, we would only minimize . Thus, the cost is minimized over a sample of the data rather than the entire data set.

When some form of online machine learning must be used, where the cost is partially minimized as each new example is seen. While online machine learning is often used when is fixed, it is most useful in the case where the distribution changes slowly over time. In neural network methods, some form of online machine learning is frequently used for finite datasets.

2.4.6 | Choosing a cost function

While it is possible to define some arbitrary, ad hoc cost function, frequently a particular cost will be used, either because it has desirable properties (such as convexity) or because it arises naturally from a particular formulation of the problem (e.g., in a probabilistic formulation the posterior probability of the model can be used as an inverse cost). Ultimately, the cost function will depend on the desired task. An overview of the three main categories of learning tasks is provided below.

2.4.7 | Learning paradigms

There are three major learning paradigms, each corresponding to a particular abstract learning task. These are supervised learning, unsupervised learning and reinforcement learning.

2.4.8 | Supervised learning

In supervised learning, we are given a set of example pairs and the aim is to find a function in the allowed class of functions that matches the examples. In other words, we wish to infer the mapping implied by the data; the cost function is related to the mismatch between our mapping and the data and it implicitly contains prior knowledge about the problem domain.

A commonly used cost is the mean-squared error, which tries to minimize the average squared error between the network's output, f(x), and the target value y over all the example pairs. When one tries to minimize this cost using gradient descent for the class of neural networks called multilayer perceptron’s, one obtains the common and well-known back-propagation algorithm for training neural networks.

Tasks that fall within the paradigm of supervised learning are pattern recognition (also known as classification) and regression (also known as function approximation). The supervised learning paradigm is also applicable to sequential

data (e.g., for speech and gesture recognition). This can be thought of as learning with a "teacher," in the form of a function that provides continuous feedback on the quality of solutions obtained thus far.

2.4.9 | Unsupervised learning

In unsupervised learning, some data is given and the cost function to be minimized, that can be any function of the data and the network's output.

The cost function is dependent on the task (what we are trying to model) and our a priori assumptions (the implicit properties of our model, its parameters and the observed variables).

As a trivial example, consider the model, where is a constant and the cost. Minimizing this cost will give us a value of that is equal to the mean of the data. The cost function can be much more complicated. Its form depends on the application: for example, in compression it could be related to the mutual information between and, whereas in statistical modeling, it could be related to the posterior probability of the model given the data. (Note that in both of those examples those quantities would be maximized rather than minimized).

Tasks that fall within the paradigm of unsupervised learning are in general estimation problems; the applications include clustering, the estimation of statistical distributions, compression and filtering.

2.4.10 | Reinforcement learning

In reinforcement learning, data are usually not given, but generated by an agent's interactions with the environment. At each point in time, the agent performs an action and the environment generates an observation and an instantaneous cost, according to some (usually unknown) dynamics. The aim is to discover a policy for selecting actions that minimizes some measure of a long-term cost; i.e., the expected cumulative cost. The environment's dynamics and the long-term cost for each policy are usually unknown, but can be estimated.

More formally, the environment is modeled as a Markov decision process (MDP) with states and actions with the following probability distributions: the instantaneous cost distribution, the observation distribution and the transition, while a policy is defined as conditional distribution over actions given the observations. Taken together, the two define a Markov chain (MC). The aim is to

discover the policy that minimizes the cost; i.e., the MC for which the cost is minimal.

ANNs are frequently used in reinforcement learning as part of the overall algorithm. Dynamic programming has been coupled with ANNs (Neuro dynamic programming) by Bertsekas and Tsitsiklis and applied to multi-dimensional nonlinear problems such as those involved in vehicle routing or natural resources management because of the ability of ANNs to mitigate losses of accuracy even when reducing the discretization grid density for numerically approximating the solution of the original control problems.

Tasks that fall within the paradigm of reinforcement learning are control problems, games and other sequential decision making tasks.

2.4.11 | Learning algorithms

Training a neural network model essentially means selecting one model from the set of allowed models (or, in a Bayesian framework, determining a distribution over the set of allowed models) that minimizes the cost criterion. There are numerous algorithms available for training neural network models; most of them can be viewed as a straightforward application of optimization theory and statistical estimation.

Most of the algorithms used in training artificial neural networks employ some form of gradient descent. This is done by simply taking the derivative of the cost function with respect to the network parameters and then changing those parameters in a gradient-related direction.

Evolutionary methods, simulated annealing, expectation-maximization, non-parametric methods and particle swarm optimization are some commonly used methods for training neural networks.

2.4.12 | Employing artificial neural networks

Perhaps the greatest advantage of ANNs is their ability to be used as an arbitrary function approximation mechanism that 'learns' from observed data. However, using them is not so straightforward and a relatively good understanding of the underlying theory is essential.

Choice of model: This will depend on the data representation and the application. Overly complex models tend to lead to problems with learning.

Learning algorithm: There is numerous trades-offs between learning algorithms. Almost any algorithm will work well with the correct hyper parameters for training on a particular fixed data set. However selecting and tuning an algorithm for training on unseen data requires a significant amount of experimentation.

Robustness: If the model, cost function and learning algorithm are selected appropriately the resulting ANN can be extremely robust.

With the correct implementation, ANNs can be used naturally in online learning and large data set applications. Their simple implementation and the existence of mostly local dependencies exhibited in the structure allows for fast, parallel implementations in hardware.

2.4.13 | Applications

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations. This is particularly useful in applications where the complexity of the data or task makes the design of such a function by hand impractical.

2.4.13.1 | Real-life applications

The tasks artificial neural networks are applied to tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction, fitness approximation and modeling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind source separation and compression.
- Robotics, including directing manipulators, Computer numerical control.

Application areas include system identification and control (vehicle control, process control, natural resources management), quantum chemistry, game-playing and decision making (backgammon, chess, poker), pattern recognition (radar systems, face identification, object recognition and more), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial

applications (automated trading systems), data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

Artificial neural networks have also been used to diagnose several cancers. An ANN based hybrid lung cancer detection system named HLND improves the accuracy of diagnosis and the speed of lung cancer radiology. These networks have also been used to diagnose prostate cancer. The diagnoses can be used to make specific models taken from a large group of patients compared to information of one given patient.

The models do not depend on assumptions about correlations of different variables. Colorectal cancer has also been predicted using the neural networks. Neural networks could predict the outcome for a patient with colorectal cancer with a lot more accuracy than the current clinical methods. After training, the networks could predict multiple patient outcomes from unrelated institutions.

2.4.13.2 | Neural networks and neuroscience

Theoretical and computational neuroscience is the field concerned with the theoretical analysis and computational modeling of biological neural systems. Since neural systems are intimately related to cognitive processes and behavior, the field is closely related to cognitive and behavioral modeling.

The aim of the field is to create models of biological neural systems in order to understand how biological systems work. To gain this understanding, neuroscientists strive to make a link between observed biological processes (data), biologically plausible mechanisms for neural processing and learning (biological neural network models) and theory (statistical learning theory and information theory).

2.4.14 | Types of models

Many models are used in the field defined at different levels of abstraction and modeling different aspects of neural systems. They range from models of the short-term behavior of individual neurons, models of how the dynamics of neural circuitry arise from interactions between individual neurons and finally to models of how behavior can arise from abstract neural modules that represent complete subsystems. These include models of the long-term, and short-term plasticity, of neural systems and their relations to learning and memory from the individual neuron to the system level.

2.4.15 | Neural network software

Neural network software is used to simulate research, develop and apply artificial neural networks, biological neural networks and in some cases a wider array of adaptive systems.

2.4.16 | Types of artificial neural networks

Artificial neural network types vary from those with only one or two layers of single direction logic, to complicated multi–input many directional feedback loop and layers. On the whole, these systems use algorithms in their programming to determine control and organization of their functions. Some may be as simple as a one neuron layer with an input and an output, and others can mimic complex systems such as dANN, which can mimic chromosomal DNA through sizes at cellular level, into artificial organisms and simulate reproduction, mutation and population sizes.

Most systems use "weights" to change the parameters of the throughput and the varying connections to the neurons. Artificial neural networks can be autonomous and learn by input from outside "teachers" or even self-teaching from written in rules.

2.4.17 | Confidence analysis of a neural network

Supervised neural networks that use an MSE cost function can use formal statistical methods to determine the confidence of the trained model. The MSE on a validation set can be used as an estimate for variance. This value can then be used to calculate the confidence interval of the output of the network, assuming a normal distribution. A confidence analysis made this way is statistically valid as long as the output probability distribution stays the same and the network is not modified.

By assigning a softmax activation function on the output layer of the neural network (or a softmax component in a component-based neural network) for categorical target variables, the outputs can be interpreted as posterior probabilities. This is very useful in classification as it gives a certainty measure on classifications.