Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Accredited by **NAAC** as '**A++**' ❖Approved by AICTE ❖ ISO 9001-2015 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. 08645 - 350200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129.

INFANT CRY ANALYSIS USING NEURAL NETWORKS IN MATLAB

A Term Paper / ProjectReport

Submitted in the partial fulfillment of the

requirements forthe award of the degree of

Bachelor of Technology

in

Department of Electronics and Communication

By

Sarvani.M -190040293

under the supervision of Dr.M.Suman

Co-Supervisor: Assoc.Prof.Dr.M.Kasi Prasad

Department of Electronics and Communication

K L E F, Green Fields,

Vaddeswaram- 522502, Guntur(Dist), Andhra Pradesh, India.

Nov, 2022

Declaration

The Project Report entitled **“ Infant Cry Analysis using Neural Network in Matlab**“ is a record of bonafide work of student studying btech final year named Sarvani.M-190040293 submitted in partial fulfillment for the award of B.Tech in Electronics and Communication to the K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

Sarvani.M -190040293

Certificate

This is to certify that the Project Report entitled **" Infant Cry Analysis using Neural Network in Matlab "** is being submitted by Sarvani.Maganti in partial fulfillment for the award of B.Tech in Electronics and Communication to the K L University is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/University/Institute..

Signature of the Co-Supervisor (If Available)

Name and Designation

Signature of the Supervisor

Name and Designation

Signature of the HOD

Signature of the External Examiner

Acknowledgement

It is great pleasure for us to express my gratitude to our honorable President Sri. Koneru Satyanarayana, for giving the opportunity and platform with facilities in accomplishing a project based on laboratory report.

I express sincere gratitude to our Coordinator and HOD for their leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor Dr.M.Suman sir and co-supervisor **Dr.M.Kasi.Prasad** sir for his novel association of ideas, encouragement, appreciation, and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

Submitted by:

Sarvani.M-190040293

Abstract

There are many reasons why an infant cries, but it is not possible to find a specific reason. Only older people can understand the reasons behind crying. An important factor in understanding an infant's crying is experience. Due to modern life scenarios, many people do not live with parents and thus lack experience in knowing the actual cause of infant crying. Therefore, applying analysis of infant crying using neural networks can help parents to know the exact reason for their infant's crying.

Dataset is collected from the playschool's and we gave the collected audios in .wav from to get it read then we computed our desired feature which is spectral spread then we stored the feature extracted, then we designed a net work with no issues and errors then fed input set and target set into that network. Where input set consist of extracted features and target set consist of categories. we included two categories hunger and pain namely 1,0 because the major emotions of the infant is hunger and pain so these should be priorly detected. After this the baby cry is take from the microphone and directed for feature extraction after computing feature extraction, we extracted data from the recorded audio, fed it into the network, and categorized the results.

As hunger and pain are out two main emotional concerns of infants , we gave the hunger set a 1 and the pain set a 0.

Index:

1.Introduction

1.1 MATLAB

This application is developed using MATLAB.
A high-performance language for technical computing is called MATLAB. In a simple-to-use interface, it mixes computation, visualization, and programming while expressing issues and solutions using well-known mathematical notation.
Common uses comprise:

- calculus and mathematics
- algorithm creation
- simulation, modeling
- data exploration, analysis, & visualization
- Engineering & scientific graphics
- Developing applications, involving creation of graphical user interfaces

Simple data in MATLAB interactive systems are arrays that do not need to be dimensioned. This makes many engineering computing tasks much faster than programming in a scalar, non-interactive language such as C. Especially when using matrix and vector formulations.

Matrix Laboratory is an abbreviation for the term MATLAB. The original purpose of MATLAB was to simplify the use of state-of-the-art matrix software developed by the eispack and linpack projects. Together they represent the state of the art in matrix computation software.

MATLAB has evolved over the years with feedback from many users. In an academic context, it is a popular teaching tool for introductory and advanced courses in mathematics, engineering and science. In the business world, MATLAB is the tool of choice for highly efficient research, development, and analysis. The

A toolbox is one type of application-specific solution available for MATLAB. Toolboxes are essential to most of her MATLAB users as they allow them to learn and use specific technologies. A toolbox is a complete collection of her MATLAB functions (M-files) that extend her MATLAB environment to address specific types of problems. There are toolboxes for many areas such as signal processing, control systems, neural networks, fuzzy logic, wavelets, and simulation.

1.2 Five major parts of the MATLAB system:

i. MATLAB language:
This is a high-level matrix with object-oriented programming features, control flow statements, functions, data structures, inputs/outputs, and inputs/outputs. / is an array

language. This allows "code-in-small" to quickly create shoddy throwaway programs, and "code-in-giant" to fully develop large and complex applications.

ii. MATLAB Working Environment:

MATLAB users or programmers work with this set of resources and tools. It contains tools for importing and exporting data and editing variables in the workspace. It also includes tools for creating, managing, debugging, and profiling M-files that are MATLAB applications.

iii. Graphics management:

displays the graphics system MATLAB. It provides general instructions for image processing, animation, 2D and 3D data visualization, and presentation graphics. It also contains low-level instructions for creating complete graphical user interfaces for MATLAB applications and completely changing the appearance of graphics.

iv. Library of math functions for MATLAB:

Contains a wide variety of computational algorithms, from simple ones such as sum, sine and cosine, to more complex ones such as matrix inversion, eigen values, Bessel functions and fast Fourier transforms.

v. An application program interface (API) for MATLAB.

You can use this library to write C and Fortran applications that communicate with MATLAB. It includes tools for reading and writing MAT files, calling his MATLAB as computational engine, and dynamically linking MATLAB functions.

1.3 DATA SET COLLECTION

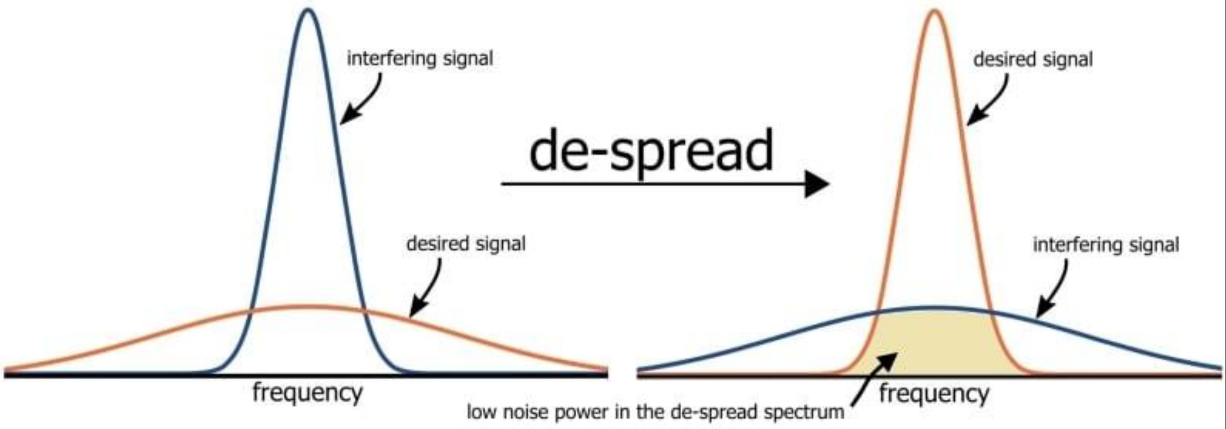
The data set gathered for the undertaking consists of the principal emotion of the toddler cry i.e.; starvation and pain. We collected the data set from present resources and from the play schools. After collecting the audio, noise elimination is completed then the characteristic extraction is done.

1.4 Feature Extracted

Spectrum Spread

Spread-spectrum techniques involve the purposeful spreading of a signal (such as an electrical, electromagnetic, or acoustic signal) in the frequency domain to produce a signal with a broader bandwidth. These methods are employed for a number of purposes, such as the establishment of secure communications, enhancing resistance to noise, jamming, and natural interference, avoiding detection, reducing power flux

density (for example, in satellite downlinks), and enabling multiple-access communications.
In other words spread spectrum technology uses pseudo-random disturbances to change the frequency of the signal being delivered. By boosting the signal's transmission bandwidth, this injection lessens the impact of signal fading, interference, and noise. Only the sender and receiver are aware of the codes used to generate the pseudo-random noise. The pseudo-random codes are utilized at the receiving end to de-spread the signals and retrieve the actual data. This makes signal transmission more secure.



1.5 Spread spectrum Advantages

The spread spectrum approach has a number of benefits that urge designers to use it into wireless communication technology.

i. **Improved signal integrity and reduced static noise:**
spread spectrum technology provides digital processing with high processing gain, making the technology immune to electromagnetic interference and noise. This gives good signal integrity with reduced static noise. The spread spectrum method significantly reduces the static noise induced in electrical equipment compared to analog wireless communication systems.

ii. **Reduced crosstalk:**
The processing gain of spread spectrum technology helps reduce crosstalk in wireless communications. In digital processing, a spread spectrum approach suppresses crosstalk. Noise below the threshold is considered a negligible error in digital signal processing.

iii. **Multipath fading immunity:**
The broadband spread spectrum approach imparts frequency diversity qualities, making signal transmission resistant to multipath fading. Signal frequencies that are several MHz apart will not decrease at the same time. By splitting the signal, the frequency hopping spread spectrum approach mitigates fading and associated communication problems.

iv. **Communications Security:**
Spread Spectrum modulates a signal in the time or frequency domain with pseudo-random noise to enable secure communications. Unlike analog radio communication systems, the random nature of the signal randomizes the

transmitted signal to ensure secure transmission. By despreading and applying the same spurious noise, the receiver recreates the signal.

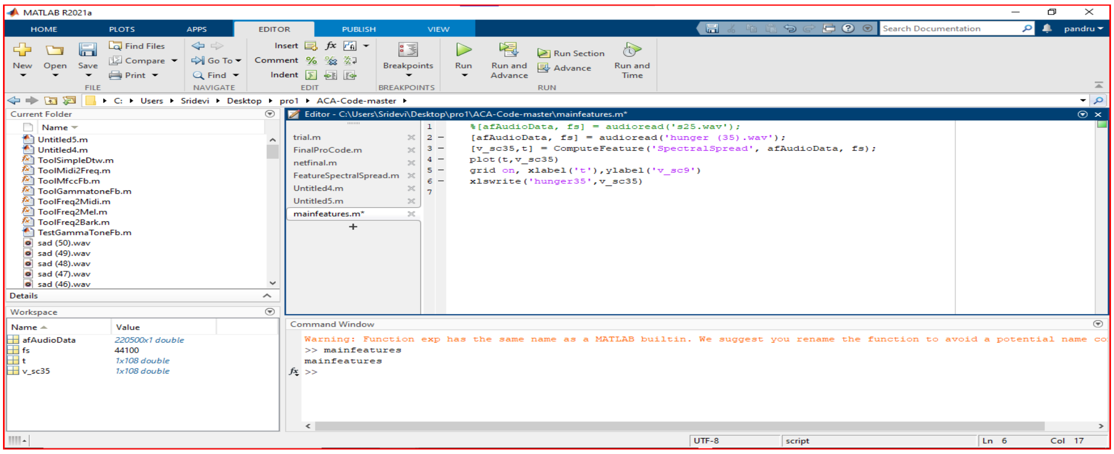
v. **Demodulation Difficulty:**
Spread spectrum approaches are difficult to demodulate because only the transmitter and receiver perceive the injected pseudo-random noise. A pseudorandom noise sequence is required to acquire the data. Without knowledge of pseudo-random noise, both decoding the signal transmission and demodulating the signal are impossible. Pseudo-random noise is long and fast, making it difficult to intercept and impossible for hackers to generate code for.

vi. **Anti-jamming:**
spread spectrum technology increases the bandwidth of the signal until the original bandwidth and the bandwidth of the injected pseudo-random noise match. This has the effect of alleviating congestion. Spread spectrum technology is augmented with pseudo-random noise to reduce noise and interference in wireless systems.

vii. **Spread Spectrum Coexistence:**
With proper planning, different spread spectrum methods can coexist in the same space without interfering with each other. Compared to non-spread spectrum systems, spread spectrum systems are less susceptible to interference. Spread-spectrum-based wireless communication has higher system capacity than analog wireless communication systems.

viii. **Longer operating range:**
spread spectrum modulated signal improves transmission power and anti-interference properties to extend operating range. Compared to analog wireless communications, spread spectrum signals can travel farther thanks to their higher transmit power capabilities.

ix. **Detection difficulty:** Spread spectrum modulated transmissions are harder to detect since their bandwidth is larger than that of traditional narrowband transmission. Wider bandwidth allows for low power transmission that is unaffected by background noise. The original signals are recovered during de-spreading when the noise frequencies are discarded.



1.4.1 Feature extraction

B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
698.8523	686.8638	737.0751	722.4694	859.9455	859.273	1442.309	1653.409	800.6588	890.4376	1057.443	741.4217	837.8296	672.3404	727.0721	1331.379	1471.984	298.865	306.2113	344.7218	352.1
628.0948	418.9986	1017.019	401.0008	471.5252	1065.133	1480.053	1286.585	1822.653	1291.275	956.2879	868.0074	635.9493	726.0783	624.0048	587.5147	683.0254	675.8172	955.8366	1469.488	690.8
259.1771	248.6038	236.883	228.2806	169.8007	279.2558	496.1094	600.1019	755.2827	546.8781	711.0968	860.2301	872.7801	1389.254	773.0778	627.037	682.8053	675.3658	753.8472	967.5654	467.5
332.7079	307.3706	367.7733	665.5761	762.931	781.6049	698.0519	748.9807	851.0962	828.1945	208.1457	572.3969	673.0642	634.7565	830.8966	1973.295	2021.121	2507.267	2172.36	2456.752	2609.
1362.148	1151.973	1213.555	345.1628	200.4166	175.148	155.5709	166.1413	252.1904	547.1936	434.9289	313.6105	308.1325	765.8018	896.1024	706.0493	567.756	603.3166	544.8442	771.8273	506.3
769.875	1019.741	1002.371	978.9482	803.0826	941.5955	1353.342	1684.725	254.8654	283.3046	440.5239	664.089	797.8917	699.4526	734.6957	798.9436	518.2772	684.7316	721.718	612.9833	498.
785.1707	1121.351	836.5696	616.8818	773.0325	840.4845	811.7859	1567.346	979.7154	752.7694	619.4352	599.1655	892.2549	684.8098	644.7234	666.4391	557.135	633.7061	529.3791	580.8506	829.0
611.568	692.3657	1311.433	654.0022	222.0769	270.759	228.1601	224.7078	249.2406	202.1644	385.6901	272.5664	350.4053	613.4939	443.8197	654.7203	518.5943	540.2128	601.0974	927.3316	943.0
611.568	692.3657	1311.433	654.0022	222.0769	270.759	228.1601	224.7078	249.2406	202.1644	385.6901	272.5664	350.4053	613.4939	443.8197	654.7203	518.5943	540.2128	601.0974	927.3316	943.0
617.49	695.3927	822.809	772.9735	832.6401	1772.378	980.5369	675.809	432.8618	309.7052	253.6304	265.6099	298.4683	670.6611	770.2018	487.391	712.2665	270.4749	187.7356	588.0939	672.9
282.6859	169.4041	352.7215	352.7402	210.4866	139.5222	258.9468	696.5464	1050.152	220.9326	228.8195	369.4658	462.3924	485.8693	609.6321	443.0668	288.3282	324.977	676.8355	427.23	929.6
1061.764	944.2449	991.088	641.3153	897.0955	815.8622	602.9436	1839.677	1408.923	845.82	659.6636	476.1156	495.6713	629.7638	493.0072	479.9357	402.475	455.0402	442.9542	394.6686	461.5
405.1997	353.6233	358.1152	480.6048	478.582	433.8157	603.3475	495.1695	1728.869	1078.136	602.2899	533.0157	525.9108	1793.354	1977.215	1629.319	1359.749	883.3077	472.9086	504.998	629.9
506.8192	385.8016	477.7973	486.4847	493.9662	542.2004	588.1687	529.7579	1135.373	1213.322	291.4882	351.9762	275.751	479.6508	618.4701	677.4754	692.8143	617.1347	889.6079	1214.669	1047.
705.8646	859.3719	972.7426	1033.703	913.4206	677.707	666.1507	801.4974	545.5504	600.6944	586.6853	610.4067	627.674	511.1355	593.492	509.8157	645.1344	545.9198	835.6676	791.2062	356.9
340.4344	352.0715	364.8712	418.4585	790.3411	739.8432	950.128	1449.483	1517.549	982.5042	1137.159	1547.051	1375.364	500.6864	424.7327	262.383	234.5754	250.7434	421.92	616.6787	508.1
545.2059	669.0978	537.2943	748.7504	959.1326	589.8204	709.235	614.2859	572.7403	484.1084	635.6942	349.8438	535.0699	473.0399	363.5971	383.5344	565.9423	232.2339	203.2511	420.0848	336.7
856.6645	491.8468	812.6299	975.6585	975.2263	567.9585	493.0219	313.3744	1093.754	881.9407	666.3818	456.6459	575.2268	772.1765	1021.93	778.5456	638.0559	575.2853	1315.05	554.5371	576.4
535.8864	334.1123	274.5699	269.8293	253.7769	395.3062	540.6501	586.4919	500.4016	445.5522	588.7022	548.7373	860.6844	821.7229	838.9277	793.6883	688.4098	581.3849	498.7548	590.5977	753.8
711.2064	753.949	673.4744	699.4798	740.9928	716.439	646.5866	598.778	593.29	500.5522	510.0036	522.8916	572.1009	555.7054	565.0933	626.11	615.226	540.152	610.0865	609.7307	620.8
582.8638	533.432	683.274	582.2562	599.4151	590.5229	598.5646	497.4336	690.8577	607.2205	776.1441	742.6651	636.5071	954.6445	724.912	813.011	781.0196	766.891	713.4256	766.7634	794.8
741.0092	415.0025	413.7773	639.844	721.5589	801.648	1914.576	1456.932	1207.184	227.7866	276.5041	295.6207	284.5073	290.2788	662.5829	549.0341	318.4861	1086.672	478.9219	487.6673	646.0
736.8135	713.4654	646.8811	624.0615	514.905	499.0282	548.9511	565.1987	549.9473	547.8354	641.458	798.8408	631.0596	957.0524	794.4925	809.3727	909.0173	645.029	777.5684	641.9494	696.8
700.8315	705.7876	693.2945	886.5194	771.2663	754.0679	830.6123	778.7915	992.8128	712.7626	642.0307	653.9415	604.264	708.856	707.2785	945.8615	648.603	684.6084	731.2549	855.0739	1293.
675.3653	864.5091	735.4004	1013.44	875.7163	829.7245	808.3009	992.4887	991.2964	801.8618	669.905	993.7083	814.2257	680.4631	984.7527	745.8918	834.8992	830.5871	771.5676	769.7367	1230.

1.4.2 Extracted feature from the samples

2.Literature Survey

2.1 Speech Processing:

The study of speech signals and signal processing techniques is called speech processing. Audio processing can be considered a specific example of digital signal processing applied to audio signals, since signals are often treated in digital representation. Capturing, modifying, storing, transmitting, and outputting audio signals are all part of audio processing. Speech synthesis is the result, speech recognition is the input.

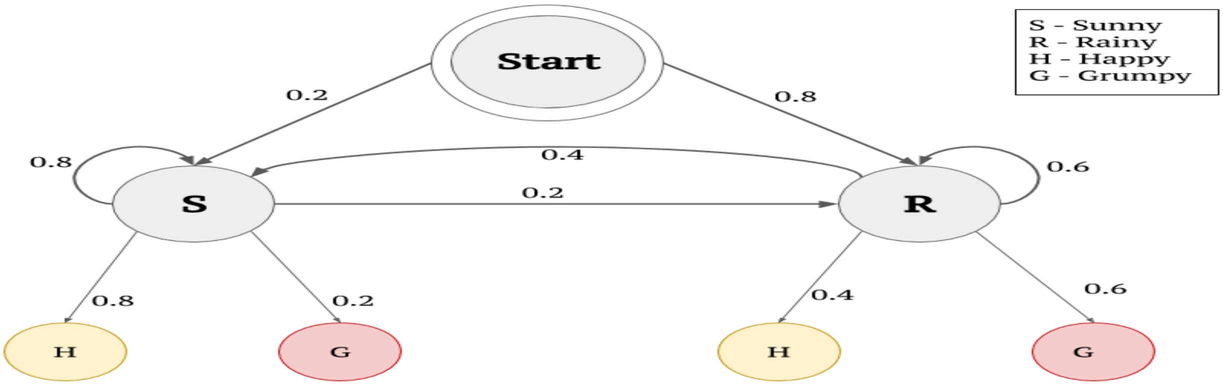
2.2 Technique:

2.2.1 Dynamic Time Wrapping:

The algorithm that compares two time sequences with different speeds for similarity is called Dynamic Time Warping (DTW). In general, DTW is a technique for determining the best possible match between two specific sequences under certain constraints and rules. The match that satisfies all constraints and rules and has the lowest cost is called the best match. The cost is computed as the sum of the absolute differences in the values of each matched index pair.

2.2.2 Hidden Markov Models:

The simplest dynamic Bayesian network can be thought of as a Hidden Markov Model. Given a list of observations y, the goal of the algorithm is to estimate the hidden variable x(t)(t). Using the Markov property, given the value of the hidden variable x, the conditional probability distribution of the hidden variable x(t) at time t depends only on the value of the hidden variable at time t (t 1) can show that The value of the hidden variable x(t) simply determines how large the observed variable y(t) grows (both at time t).



2.2.3 Artificial Neural Networks:

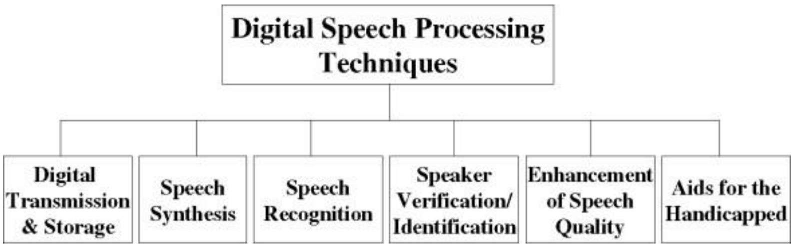
Artificial neurons, groups of interconnected units or nodes, are much like neurons in the biological brain and form the basis of artificial neural networks (ANNs). Each connection can send a signal from one artificial neuron to another, just like synapses in the human brain. After processing the signal, the artificial neuron can give the signal to other artificial neurons connected to it. In traditional ANN implementations, the output of each artificial neuron is computed by a nonlinear function of the sum of its inputs, and the signals at the connections between artificial neurons are real.

2.2.4 Other Voice Applications:

This diagram shows various voice communication applications. In addition to the three areas of transmission/storage, speech synthesis, and speech recognition, many other areas, such as speaker identification, speech signal quality enhancement, and assistance for the visually impaired, include digital speech processing technology as an integral part. It is A system in which a time signal containing speech is processed using DSP methods is represented by the block diagram in Figure.

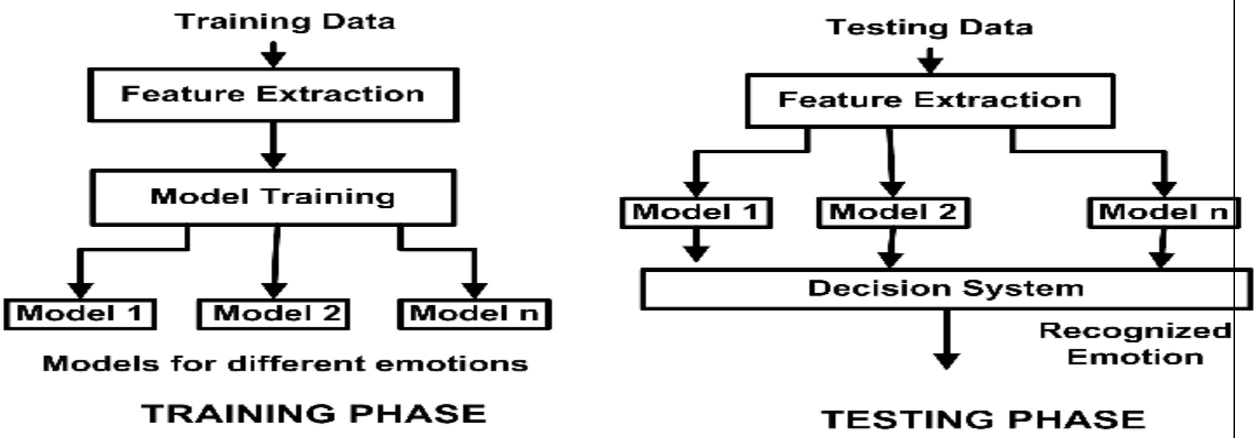
This graph demonstrates the idea that a DSP method can modify the captured audio signal essentially indefinitely. Again, manipulating and modifying an audio signal typically involves transforming the audio signal into another representation (based on knowledge of audio production and perception) and performing operations on that representation through additional digital computations. , is done by transforming to the waveform domain using D/A converter.

Speech enhancement is the primary application of removing or reducing background noise, echo, or reverberation picked up by a microphone along with the target speech signal. The goal of speech enhancement systems in human-to-human communication is to make speech more intelligible and natural. But in practice, the best achieved so far is a less sensible conversation that essentially preserves but does not improve the intelligibility of loud speech. However, it has been successful in improving the usefulness of distorted speech signals for additional processing as components of speech encoders, synthesizers, or recognizers.



2.3 Types of speech recognition:

Speech recognition comes in two different types. Speaker-dependent is the first, while speaker-independent is the second. While speaker-independent software is more frequently utilized in telephone applications, speaker-dependent software is frequently employed for dictationsoftware.



In a manner similar to voice recognition, speaker-dependent software functions by learningthe distinct qualities of a single person's speech. To for the programme to understand how new users speak, they must first "train" it by speaking to it. This frequently entails that before using speech recognition software, users must read a few pages of text to the computer.

There is no need for training because speaker-independent software is made to detect anyone's voice. Since businesses cannot require callers to read pages of text before utilising the system, it is the only practical alternative for applications like interactive voice response systems.Software that is speaker-independent is typically less accurate than software that is speaker- dependent, which is a drawback.

In order to address this issue, speaker independent speech recognition engines typically restrict the grammars they employ. The speech engine is more likely to correctly understand what aspeaker said by using a smaller set of recognized terms.

For most IVR systems and any application where a sizable number of users will be using thesame system, speaker-independent software is therefore perfect. In dictation software, where just one person will utilise the system and a big grammar is required, speaker dependent software is more commonly employed.

All of our speech software is powered by the speaker-independent Lumen Vox Speech Engine. It is not the same as voice recognition software, it cannot recognize an infinite number of words at once, and it is not dictation software. It is made to recognize certain information, primarilyfrom callers who enter it into an IVR on a phone. It functions effectively in applications like call routing, auto-attendants, and other systems where developers anticipate the language a speaker would use.

We employ hundreds of hours of audio that has been transcribed to create a language model in order to create it. This database explains to our speech engine what mathematical soundsseem like because arithmetic is the only language that computers can actually understand.

The Engine can distinguish a wide range of voices since the audio we use to create the models has hundreds of speakers. It is speaker-independent due to this.The Engine transforms the speaker's audio into a mathematical representation when it gets input from a voice application andcontrasts it with its internal models. The Engine can then compare these sounds to the words listedin the grammar of the voice application after gaining an understanding of the sounds that make upthe audio.

This procedure is not exact. The Speech Engine can never be certain of what the speaker saidsince there are so many small differences in how words are pronounced. For instance, if the audio quality is poor, even humans can never be certain of what someone said to them. Think about how tough it is to tell the difference between the letters "t" and "b" while spelling a word.

Our voice recognition program uses a probabilistic approach to deal with this ambiguity.The Speech Engine provides a confidence score for every piece of audio it tries to identify, similar to how a public opinion poll has a margin of error for a particular confidence threshold. The likelihood that the Engine's recognition result corresponds to what the speaker stated is indicated by this score.

2.4 Speech recognition (Independent):

To enable programmes that use voice recognition to process the input, phonemes from the provided audio are extracted, translated to ASCII characters, and then formulated into words for computer systems that are speaker-independent. To determine the most likely word spoken, mathematical algorithms and models are applied. These models compare spoken words to recognised word models and choose the one that has the best chance of being the right word. Largequantities of training data are used to build the models in order to determine the "highest likelihood." The Hidden Markov Model is a particular kind of statistical model (HMM).

A finite-state Markov model and a number of output distributions define an HMM. The twoforms of variability, temporal variability and spectral variability, both capture the essence of voice recognition. While the latter is modeled by the parameters in the output distribution, the former is modeled by the transition parameters in the Markov chain.

HMMs are suitable for continuous voice recognition because they are built on a solid probabilistic foundation and provide an integrated framework for simultaneously tackling the segmentation and classification problem. A "natural" interface to the machine, one where the flowof conversation is not disrupted by forced pausing, is difficult for users of other systems where middle silence detection (a pause of some unintelligible utterance in the middle of speech) is difficult. In these systems, the user is asked to utter each word separately and wait for the system torecognize.

It has become simple to recognize speech regardless of the speaker's accent by viewing speech as an ordered collection of phonemes. Although independent speech recognition systemsuse massive samples to build their models, training is still necessary.

Dynamic time wrapping is a pattern-matching method that is also utilised to HMM. By adding the changes between speech frames, this approach compares the pre-processed speech to areference template. Stretching and compressing are used to correct some of the words that are outof alignment with the provided template.

Use of neural networks is a relatively contemporary method of independent speech recognition. HMM technology, as previously mentioned, operates by assuming specific things aboutthe structure of speech recognition and then estimating system parameters as though the assumptions were true. If the presumptions are wrong, this method could fail. Such presumptions are not required by the neural network approach. This method makes use of a distributed representation of straightforward nodes, the connections of which have been trained to identify speech. As opposed to HMMs, which disperse knowledge or constraints across many simple computing units rather than encoding it in discrete rules or procedures. Uncertainty is portrayed bythe pattern of activity in several units rather than the unlikelihood of a single unit.

2.5 Speech recognition (Dependent):

For the purpose of supporting a single speaker, a speaker-dependent system is created. These systems are typically more accurate, less expensive to buy, and easier to construct. The system can operate much more quickly and precisely after being trained to recognise each user'spronunciations, inflections, and accents. Users must take part in training sessions to help the computer "learn" to identify their voice. After that, the computer creates a voice profile that is tailored to the necessary training.

Like other speaker-dependent recognition systems, triphones, multiple words, and phonemes are matched in order to function. For systems with bigger vocabularies, up to ten thousand words, the phoneme/multi-word technique is generally utilised. With medium to largevocabularies and either isolated or connected word recognition, speaker-dependent systems typically perform well.

Although these speaker-dependent systems still have room for improvement and are far from ideal, they should not only be used for dictation and word processing. There are numerous possibilities to investigate, and it is likely that this will be used and adopted widely in the years to come.

Speaker identification:

Understanding what the user is requesting and who the user is important when numerous family members utilize digital home assistants. The latter is crucial to accurately responding to questions like, "When is my next appointment?" The system must carry out utterance-by-utterance speaker recognition to accomplish this. Speaker identification algorithms can run locally on the device or remotely on a server, and they can be text-dependent (usually based on the wake-up keyword) or text-independent. Typically, a registration procedure is required before the assistant may link speech to a user profile. It is possible to implement enrolment by requesting a user’s identify and a few sample sentences up front.

Natural Language Processing:

The "real" concept of humans seeking to interact with computers is Natural Language Understanding; these are computer programs that can understand the task users are attemptingto perform without the need to employ the narrow vocabulary required by speech recognition programs. Instead, then concentrating on phonemes, NLU examines the context of the speech, much like a human would. Though not fully integrated with voice recognition now, this area of research is thought to form the basis for speech recognition in the future.

2.6 Speech Recognition Process:

- Speech or Audio
- Speech or Audio Preprocessing
- Feature Extraction
- Speech Classification
- Recognition

i. **Speech:**

There are many devices and software programs that can record human voice. The sound produced is highly influenced by the sound environment and the technology used. While quite inconvenient, background noise and room reverberation can add to conversations.

ii. **Audio Preprocessing:**

Audio Preprocessing is the answer to the above problem. This has a significant impact on eliminating sources of insignificant variance. Dereverberation, echo cancellation, windowing, noise filtering, and smoothing are common speech preprocessing techniques, all of which significantly improve speech recognition accuracy.

iii. **Extraction of Feature:**

Each person speaks differently and has a different intonation. This is due to various features ingrained in their language. In theory, you should be able to recognize speech with the theoretical waveform. Given the enormous diversity of languages, there is an urgent need to perform some form of feature extraction to reduce variability.

The following section illustrates a few of today's most popular feature extraction techniques

LPC (Linear Predictive Coding):

This is one of the most effective speech analysis techniques for coding high quality speech at low bit rates. The basic premise of this approach is that a given audio sample at a given time can be loosely represented as a linear combination of past audio samples. Digital signals are therefore compressed for efficient transmission

and storage. The goal of LPC is to reduce the sum of squared times between the original and estimated speech over a finite period of time. It can also be applied to provide a specific set of predictive coefficients. Gain (G) is another important metric.

MFCC (Mel-Frequency Cepstal Coefficients):

This is how feature extraction is often done. It is based on preliminary frequency ranges using the Mel scale, which is based on the scale of the human ear. It belongs to the category of frequency domain features and is therefore more accurate than time domain features. The most obvious obstacle is sensitivity to noise, as it is highly dependent on spectral shape. Speech also contains non-periodic material, but techniques that take advantage of the periodicity of the speech signal can be used to circumvent this problem.

iv. **Speech Classification:**
Speech classification refers to a set of tasks or problems for a program to automatically classify input utterances or speech segments into categories such as: B. Such as voice command recognition (multi-class), voice activity recognition (binary or multi-class), and audio sensory classification (typically multi-class).

Speech Command Recognition is the task of classifying input audio samples into discrete sets. class. It is a subset of Automatic Speech Recognition (ASR), also known as keyword detection, where the model always analyzes speech patterns to recognize a specific class of "commands". Recognition of these commands allows the system to perform certain actions. Often, the goal of command recognition models is to be small and efficient so that they can be deployed in low-power sensors and remain active for long periods of time.

Voice Activity Detection (VAD), also known as voice activity detection or speech recognition, is responsible for predicting which parts of the input audio contain speech and background noise. This is an essential first step for a wide variety of speech-based applications, including automatic speech recognition. Its purpose is to specify which samples are sent to the model and when to close the microphone.

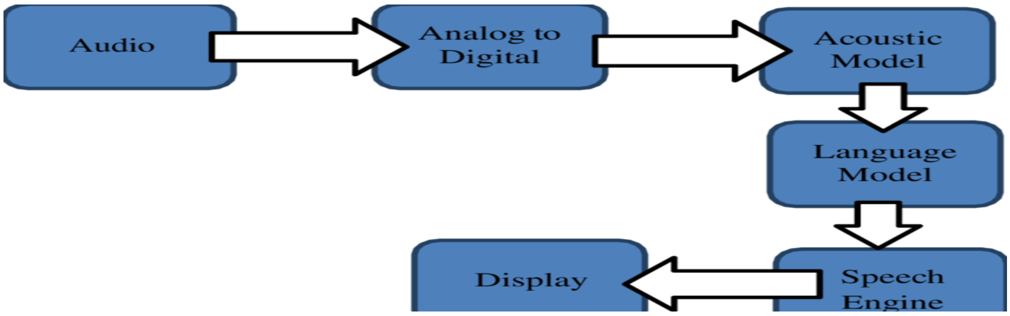
Spoken Language Identification (Lang ID), also known as Spoken Language Recognition, is the task of automatically recognizing the language of speech. This typically acts as a precondition for ASR and determines which ASR model to activate based on language.

v. Recognition:

Speech Recognition is the final stage completed after the above four steps of speech recording, speech preprocessing, feature extraction and speech classification. If all the above methods are completed successfully, you can use 3 methods to recognize speech.

1. Acoustic-speech approach
2. Pattern Recognition Approach
3. Artificial Intelligence Approach

This uses both pattern recognition and auditory speech approaches. This method uses a system built using neural networks to classify and identify sounds. Speech recognition is a particularly powerful application of neural networks. Various networks are used for this task. Speech recognition uses RNNs, LSTMs, deep neural networks, and hybrid HMM-LSTMs.



Speech Recognition Process

2.7 Neural Networks

For more than a decade, a technique called "deep learning" has given rise to some of the most powerful artificial brain frameworks, including discourse recognition in cell phones and Google's latest program interpreter.

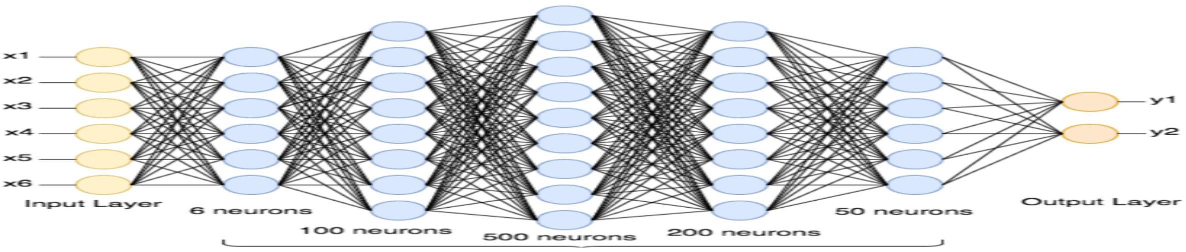
Deep learning is really just a new nickname for neural networks. Neural networks are artificial intelligence techniques that have been used for over 70 years. His two University of Chicago scholars, Warren McCullough and Walter Pitts, moved to MIT in 1952 and became a founding member of what is commonly called the first Cognitive Sciences Division, and in 1944 first proposed neural networks. .

Neural networks were an important area of research in both neuroscience and computer science until they were said to have been wiped out in 1969 by MIT mathematicians Marvin Minsky and Seymour Papert.

2.7.1 What is neural network?

A system of hardware and/or software that is based on how neurons in the human brain work is known as an artificial neural network (ANN) in the field of information technology (IT). Artificial neural networks (ANNs), also referred to as neural networks, are a sort of deep learning technology that falls under the artificial intelligence (AI) umbrella.

These technologies are often applied in the business world to solve complex signal processing or pattern recognition problems. Since 2000, handwriting recognition for check processing, speech-to-text transcription, data analysis for oil exploration, weather forecasting, and facial identification have all been significant commercial applications.



2.7.2 How do neural networks work?

ANNs often use many parallel processors arranged in layers. Raw input data is received in the first layer, similar to the optic nerve in human image processing. Each subsequent layer receives the output from the previous layer instead of the raw input, just as neurons far from the optic nerve receive signals from neurons closer to it. The output of the system is produced at the last stage.

Each processing node has its own limited body of knowledge. This includes what it saw and the rules it developed or originally coded. Each node at level n is connected to nodes at level n-1 that act as its inputs, and nodes at level n+1 that provide input data to these nodes. This is because the levels are densely connected. The output layer can contain one or more nodes that can read the generated solution.

Artificial Neural Networks are known for their adaptability. That is, it changes as you gain insights from the initial training and acquire additional data from subsequent runs. The most basic learning model is based on the concept of input stream weighting, where each node assigns a value to the importance of input data from each ancestor. Inputs are given higher weight to help provide an accurate answer.

2.7.3 How neural networks learn?

A vast amount of data is first used to train or feed an ANN. Giving the network input and specifying the desired output constitute training. For instance, the initial training could consist of a collection of images featuring the faces of actors, non-actors, masks, statues, and animals in order to create a network that recognize the faces of people perfectly . Each input has its corresponding identification, such as the names of the actors or information indicating that they are not actors or humans. By providing the responses, the model can modify its internal weightings and improve how well it performs.

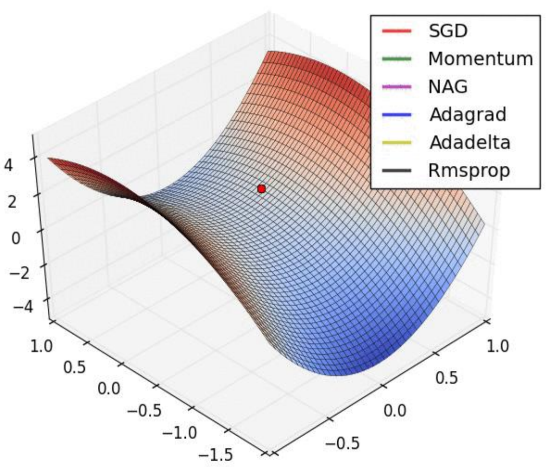
For instance, The training software believes the current input image is truly a Tom Cruise image, despite claims to the contrary from nodes D and E indicating it is a Tom Holland image. If it is determined that the image is a Holland image, the E D input is given less weight while the A, B, and C input are given greater weight.

Neural networks employ a number of concepts while creating the rules and making decisions, i.e., based on information from the previous layer when determining what to send to the next tier. These consist of Bayesian methods, genetic algorithms, gradient-based training, and fuzzy logic. The links between the objects in the data being modeled may be described to them in some simple terms.

2.7.4 Several principles of Neural networks:

Gradient Based

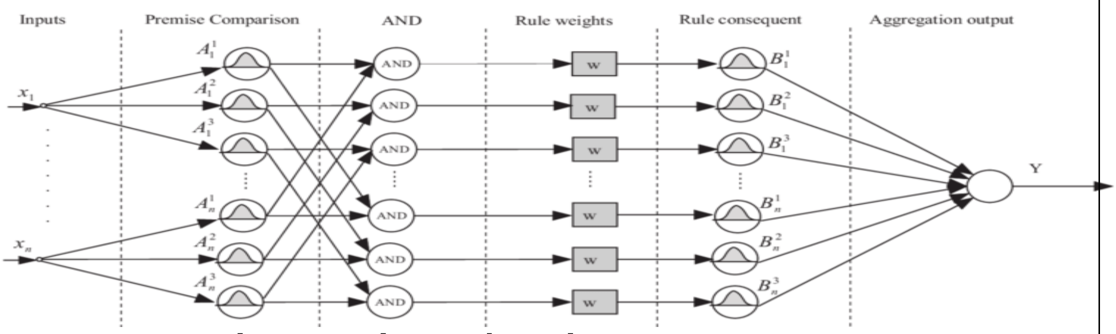
We try to approximate an input-output function by randomly initializing the network's parameters first, and then gradually updating them to find the optimal configuration of these parameters by minimizing a loss function that, in most cases, is non convex in nature (there are typically multiple local minima rather than a single global minima).
As a result, training a neural network is a nondeterministic combinatorial optimization problem because we cannot be sure that the final result will be accurate.
The following methods have been suggested to tackle this situation: SGD, momentum based, NAG, RMSprop, and ADAM. They are all variations of the classical gradient descent algorithm, and ADAM, which combines RMSprop and momentum, is thought to be the state of the art at the present.
In other words, gradient only records the change in each weight relative to the error change. A gradient is similar to a function's slope in terms of conceptualization. The steeper the slope and the quicker a model can learn, the higher the gradient. However, the model stops learning if the slope is zero.



2.7.5 Fuzzy Logic

several justifications for employing fuzzy logic in neural networks:

i. The weights of neural networks, derived from fuzzy sets, are typically defined using fuzzy logic.
ii. When applying crisp values is not possible, fuzzy values are used.(The value of a crisp set is either 0 or 1. The value between 0 and 1 that includes both 0 and 1 is defined as a fuzzy set.)
iii. We are aware that learning and training make neural networks more resilient to unforeseen events. Crisp values would not be as useful at that time as fuzzy values.
iv. The values must not be precise when using fuzzy logic in neural networks so that parallel processing is possible.



Fuzzy logic neural network topology

2.7.5.1 Several instances of neurally trained fuzzy systems

Numerous commercial applications use fuzzy systems that have undergone neural training.

i. water- and energy-saving device, the German AEG Corporation uses a fuzzy control system that has undergone neural training. There are 157 fuzzy rules in total.
ii. Trainable fuzzy systems have been developed by Ford Motor Company to regulate vehicle idle speed.

2.7.6 Types of Neural Networks

The number of layers between the input and output, are the model's "hidden layers," are commonly used to describe the depth of neural networks. Because of this, the terms "neural network" and "deep learning" are often used similarly. They can alternatively be defined in terms of the model's hidden node count are the number of inputs and outputs that each node possess. Different types of information can be propagated forward and backward among levels using modifications to the conventional neural network architecture.

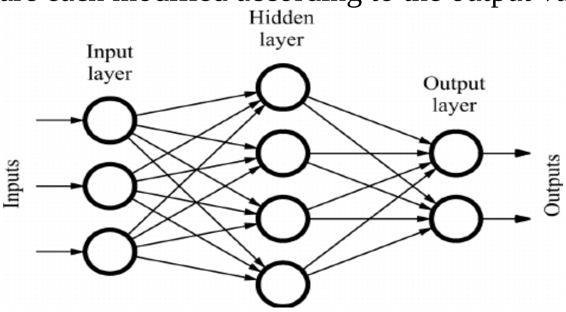
2.7.6.1 Feed Forward Neural Network

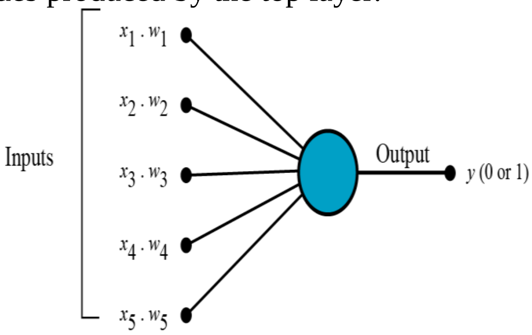
One kind of artificial neural network is a feed forward neural network in which there is no cycle in the connections between the nodes. Since input is only processed in uni direction, the feed forward model is the simplest type of neural network. Even if the data may pass across several nodes, it always proceeds forward and never backward.

2.7.6.2 What is the process of a feedforward neural network?

A single layer perceptron is a common example of a feedforward neural network in its most basic configuration. In this model, a number of inputs are introduced into layers and multiplied by weights. Then the weighted input values are added to the total. The generated value is often 1, and if the sum of the values is below the threshold, the output value is 1. The threshold is usually set to zero. For classification tasks, single-layer perceptrons are important feed-forward models for neural networks.

Single-layer perceptrons may also incorporate artificial intelligence capabilities. A neural network can compare a node's output to a desired value using a property called the delta rule. This allows the network to train weights to produce more accurate output values. This learning and training process leads to gradient descent. The process of updating weights in multi-layer perceptrons is almost identical, but is more formally called backpropagation. In such situations, the hidden layers of the network are each modified according to the output values produced by the top layer.





2.7.6.3 Recurrent Neural Network

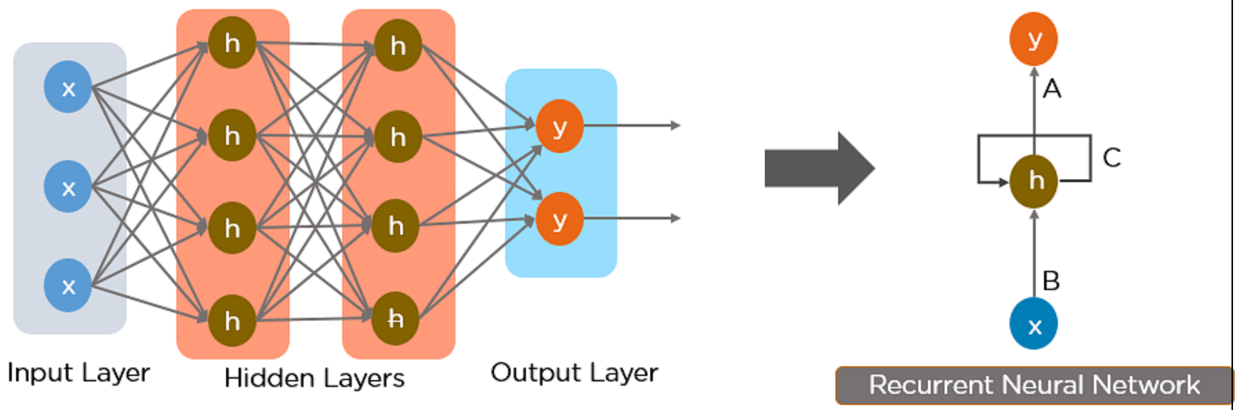
RNNs work on the concept that the output of each layer is stored and fed back to the input of the system to estimate the output of that layer.

The nodes of multiple layers of neural networks are compressed to create a single layer of iterative neural networks. The network parameters are A, B, and C.

There were several problems with feedforward neural networks that led to the development of RNNs.

- can't deal with consecutive data
- merely considers current input
- can not remember earlier input

RNN offers a remedy for these problems. RNNs can handle sequential data and accept both present-day and historical input. Previous inputs can be recalled by the RNN thanks to its internal memory.



2.7.6.4 Working of Recurrent Neural Network:

The neural network's input is received by the input layer "x" which processes it before sending it to the middle layer.
Multiple hidden layers with unique activation functions, weights, and biases may make up the middle layer "h." Recurrent neural networks can be used with neural networks without memory, meaning that the various parameters of different hidden layers are not influenced by the previous layer.
So that each hidden layer has the same characteristics, the recurrent neural network will standardise the various activation functions, weights, and biases. Then, it will build one hidden layer and loop over it as many times as necessary rather than several hidden layers.

2.7.6.5 Convolutional Neural Network(CNNs)

A deep learning network architecture also called as convolutional neural network (CNN) which learns from data directly, doing away with the requirement for human features extraction.CNNs are very helpful for recognizing objects, faces, and scenes in photos by looking for patterns in the images. For categorizing non-image data, such as audio, time series, and signal data, they can be highly useful

CNNs are widely used in computer vision and object recognition applications, including those for self-driving cars and facial recognition.

2.7.6.6 Working Process of CNNs ?

Tens or even hundreds of layers can be present in a convolutional neural network, and each layer can be trained to recognize various aspects of an image. Each training image is subjected to filters at different resolutions, and as a result of every convolved image is utilized as the input to the following layer. Beginning with relatively basic properties like brightness and borders, the filters can get more complicated until they reach characteristics that specifically identify the object.

A CNN is made up of an input layer, an output layer, and many hidden layers in between, similar to other neural networks. These layers carry out operations on the data in order to discover unique characteristics of that data.

Three most used layers.

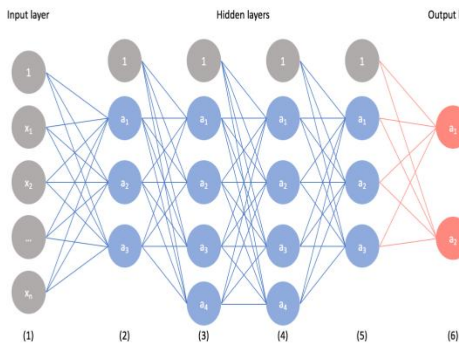
- Convolution
- Activation or ReLU
- Pooling

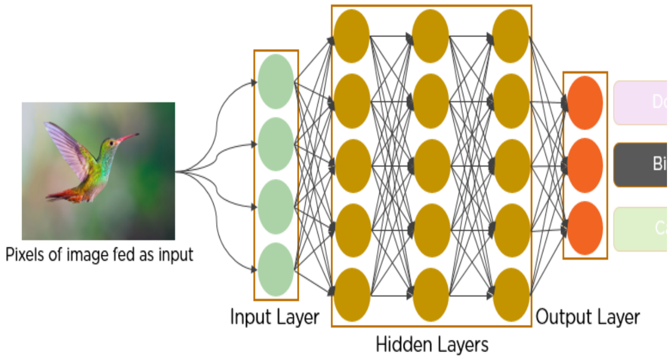
i. **Convolution:** runs a series of convolutional filters through the input images, activating different aspects of the images with each filter.

ii. **Rectified linear unit (ReLU):** which maintains positive values while translating negative values to zero, enables quicker and more efficient training. Due to the fact that only the activated features are carried over to the following layer, this is frequently referred to as activation.

iii. **Pooling:** reduces the number of parameters the network needs to learn while doing nonlinear down sampling, which simplifies the output.

Each layer learns to recognise various traits as these procedures are repeated across tens or hundreds of levels.





3 Deep network designer:

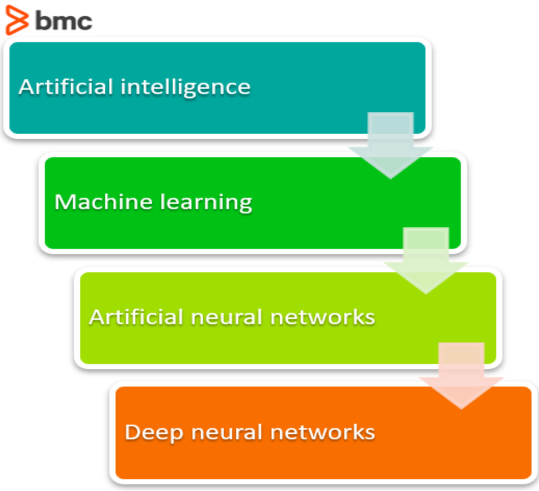
The Profound Organization Originator application allows you to assemble, envision,alter, and train profound learning organizations. Utilizing this application, you can:

- Build, import, edit, and combine networks.
- Load pretrained networks and alter them for move learning.
- View and alter layer properties and add new layers and associations.
- Dissect the organization to guarantee that the organization engineering is characterizedaccurately, and distinguish issues prior to preparing.
- Import and envision datastores and picture information for preparing and approval.
- Apply increases to picture arrangement preparing information and envision theappropriation of the class names.
- Train organizations and screen preparing with plots of exactness, misfortune, and approvalmeasurements.
- Send out prepared organizations to the work area or to Simulink.
- Produce MATLAB code for building and preparing networks and make tests forhyperparameter tuning utilizing Analysis Supervisor.

3.1 What is a Deep Neural Network?

Profound brain networks offer a ton of significant worth to analysts, especially inexpanding exactness of an AI model. The profound net part of a ML model truly got A.I.

At its easiest, a brain network with some degree of intricacy, as a rule no less than twolayers, qualifies as a profound brain organization (DNN), or profound net for short. Profound nets process information in complex ways by utilizing modern number related demonstrating.



Evolution

To begin with, AI needed to get created. ML is a system to mechanize (through calculations)measurable models, similar to a direct relapse model, to get better at making expectations. The fact that makes forecasts about something makes a model a solitary model. Those expectations are made with some exactness. A model that learns AI takes generally itsterrible expectations and changes the loads inside the model to make a model that commitsless errors.

The learning part of making models brought forth the advancement of fake brain organizations. ANNs use the secret layer as a spot to store and assess how critical one of thesources of info is to the result. The secret layer stores data in regards to the info's significance, and it additionally makes relationship between the significance of blends of sources of info.

Profound brain nets, then, benefit from the ANN part. They say, on the off chance that that functions admirably at working on a model — in light of the fact that every hub in the secret layer makes the two affiliations and grades significance of the contribution to deciding the result — why not stack increasingly more of these upon one another and benefit considerablymore from the secret layer?

Thus, the profound net has various secret layers. 'Profound' alludes to a model's layers beingdifferent layers profound.
Improving Accuracy: The black box Problem –

Profound nets permit a model's exhibition to increment in exactness. They permit amodel to take a bunch of sources of info and give a result. The utilization of a profound net isessentially as straightforward as reordering a line of code for each layer. It doesn't make any difference which ML stage you use; guiding the model to involve two or 2,000 hubs in each layer is pretty much as straightforward as composing the characters 2 or 2000.

The Profound Net permits a model to make speculations all alone and afterward store those speculations in a secret layer, the black box. The black box is difficult to research.Regardless of whether the qualities in the black box are known, they don't exist inside a system for understanding.

3.2 Deep learning for signal data:

Profound learning for signal information requires additional means when contrastedwith applying profound learning or AI to different informational collections. Great quality signinformation is difficult to acquire and has such an uproar and changeability. Wideband clamour, nerves, and twists are only a couple of the undesirable qualities tracked down in most sign information.

Similarly as with all profound learning projects, and particularly for signal information, your prosperity will quite often rely heavily on the amount of information youpossess and the computational force of your machine, so a decent profound learning workstation is strongly suggested.

To sidestep utilizing profound learning, a careful comprehension of sign information and sign handling will be required to utilize AI procedures which depends on less informationthan profound learning.

3.3 Deep learning work flow:

i. Right off the bat, the interaction would include putting away, perusing and pre- handling the information. This will likewise include separating and changing highlightsand parting into preparing and test sets. In the event that you are wanting to utilize a managed learning calculation, the information will require marking.

ii. Picturing the information will be critical to distinguishing the kind of pre-handling andelement extraction strategies that will be required. For signal handling, imagining is demanded in the investment, recurrence and time-recurrence areas for legitimate investigation.

iii. When the information has been imagined, it will be important to change and concentrate highlights from the information, for example, tops, change focuses andsignal examples.

Before the coming of AI or profound learning, traditional models for timeseries examination were utilized since signals have a period explicit space.

3.3.1 Classical Time Series Analysis

Visual review of time series, seeing change over the long haul,assessing pinnacles and box.

3.3.2 Frequency Domain Analysis

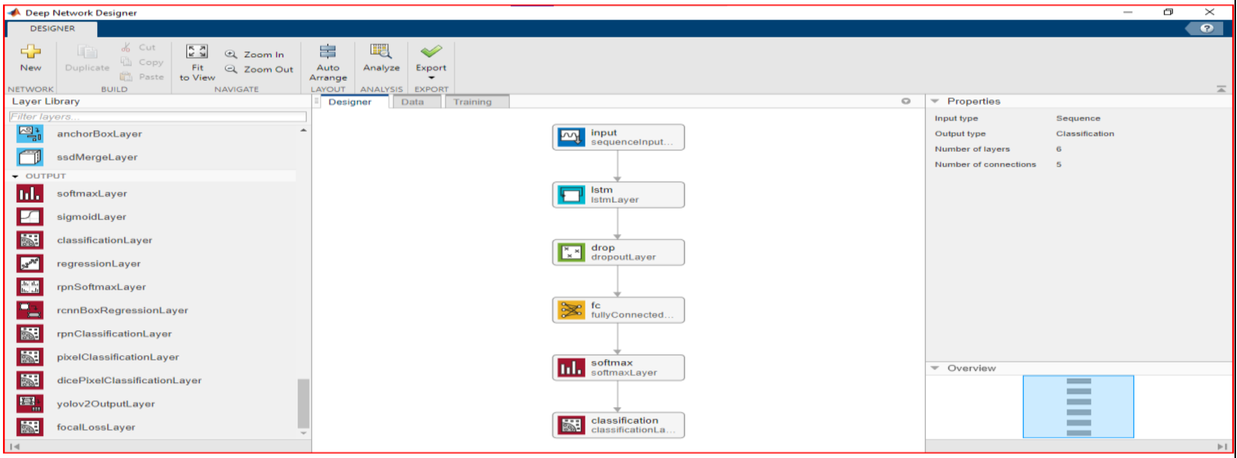
As per MathWorks, Recurrence Space Investigation is one of the critical parts of Sign Handling. It is utilized in regions like Correspondences, Topography, Remote Detecting, and PictureHandling. Time Space Examination shows a sign's energy conveyed after some time while a recurrence space portrayal remembers data for the stage shift that should be applied to every recurrence part to recuperate the first time signal with a blend of all the singular recurrence parts. A sign is changed among time and recurrence spaces utilizing numerical administrators called a "Change". Two renowned instances ofthis are Quick Fourier Change (FFT) and the Discrete FourierChange (DFT).
Long short-term Memory problem:

Human Movement Acknowledgment (HAR) has been building up momentum latelywith the appearance of propelling human PC cooperations. It has genuine applications in enterprises going from medical care, wellness, gaming, military and route.

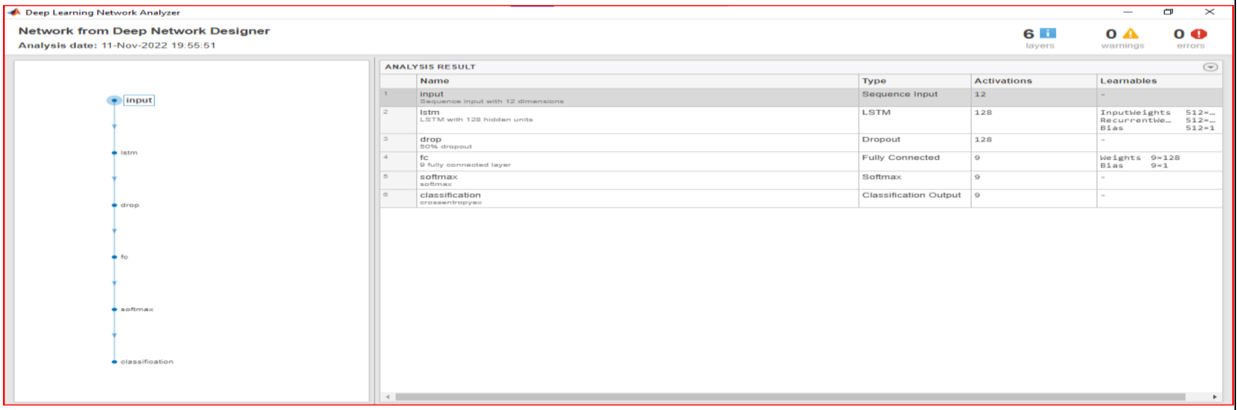
Sensor based HAR (wearables that are joined to a human body and humanaction is converted into explicit sensor signal examples that can be portioned and recognized)

External device HAR Profound Learning procedures have been utilized to beat the inadequacies of AI methods that follow heuristics framed by the client. Profound Learning techniques that can consequently extricate highlights, scale better for additional intricate undertakings. Sensor information is developing at a quick speed (eg: Apple Watch, Fitbit, walkerfollowing and so on) and how much information created is adequate for profound learning techniques to learn and produce more exact outcomes.

Intermittent Brain Organizations are a reasonable decision for signal information as it intrinsically has a period part, in this way a consecutive part. This Paper: Profound Intermittent Brain Organizations for Human Action Acknowledgment frames some LSTM based Profound RNN's tofabricate HAR models for grouping exercises planned from variable length input successions.
Organizations for Human Action Acknowledgment frames some LSTM based Profound RNN's tofabricate HAR models for grouping exercises planned from variable length input successions.



Layers of deep network designer



3.4 Layers of Deep Network Designer:

3.4.1 Sequence Input Layer:

- Size of input, specified as a positive integer or vector of positive integers.

- For vector sequence inputs, InputSize is a scalar equal to the number of features.
- For input one-dimensional image sequences, InputSize is a two-element vector [h c]. where h is the height of the image and c is the number of image channels.
- For input 2D image sequences, InputSize is a 3-element vector [h w c]. where h is the image height, w is the image width, and c is the number of image channels.
- For input 3D image sequences, InputSize is a four-element vector [h w d c]. where h is the image height, w is the image width, d is the image depth, and c is the number of image channels.
- Use the MinLength property to specify the minimum sequence length of the input data.

Length of Minimum Data:

Minimum sequence length of input data. Specified as a positive integer. When training or predicting a network, if the input data time step is less than MinLength, the software throws an error.

When creating a network that downsamples data in the time dimension, you need to ensure that the network supports training data and data for prediction. Some deep learning levels require the input to have a minimum sequence length. For example, a one-dimensional convolutional layer requires the input to have at least as many time steps as the filter size.

As time series of sequence data propagate through the network, the length of the sequences can change. For example, downsampling operations such as 1-D convolution can output data in fewer time steps than the input. This means that the downsampling operation can introduce errors at later layers in the network because the data burst length is shorter than the minimum length required by the layer.

When training or building a network, the software automatically checks whether sequences of length 1 can be propagated through the network. Some networks may not support sequences of length 1, but can successfully propagate longer sequences. To check that a network supports propagating your training and expected prediction data, set the MinLength property to a value less than or equal to the minimum length of your data and the expected minimum length of your prediction data.

3.4.2 Istm Layer:

LSTM layers learn long-term dependencies between time steps in temporal and sequence data.

This layer performs additional interactions that help improve gradient flow over long sequences during training.

Number of Hidden Units:

- This property is read-only.
- Number of hidden units (hidden size). Specified as a positive integer.
- The number of hidden units corresponds to the amount of information stored between time steps (hidden states). Hidden states can contain information from all previous time steps, regardless of sequence length. Too many hidden units can cause the layer to overfit the training data. This value can vary from tens to thousands.
- Hidden states do not limit the number of time steps processed in one iteration. Use the SequenceLength training option to split the sequence into smaller sequences and use the trainNetwork function.
- The layer outputs data in NumHiddenUnits channels.

Input Size:

This property is read-only.

Input size specified as a positive integer or 'auto'. If InputSize is "auto", the input size is automatically assigned during training.

Data type: double |or char

Output Size:

This property is read-only.

Output mode specified as one of:

'sequence' - Output the complete sequence.

'last' - Returns the last time step of the sequence.

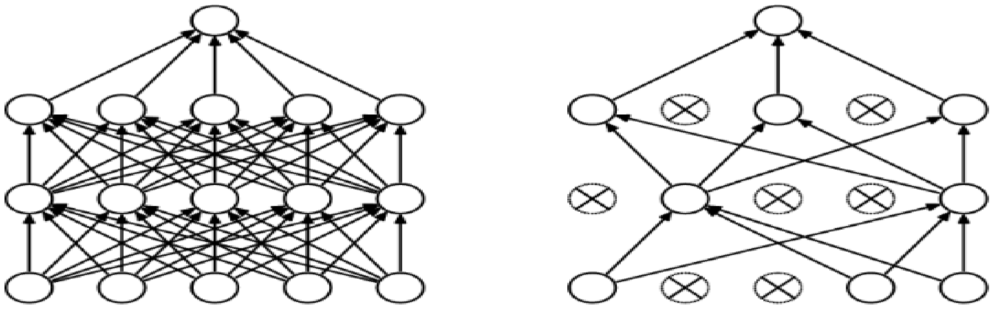
3.4.3 Drop Out Layer:

Deep neural networks have a variety of architectures, some superficial and some very deep, to generalize a given data set. However, in this quest to learn different features from datasets, we sometimes learn statistical noise in datasets. This definitely improves the model's performance on the training dataset, but fails significantly on the new data points (the test dataset). This is the problem of over fitting. There are various regularization techniques that penalize network weights to address this problem, but they weren't enough.

The best way to over fit or regularize a fixed size model is to take the average prediction from all possible settings of the parameters and aggregate the final output. However, this becomes too computationally intensive and not suitable for real-time inference/prediction.

Another method is inspired by ensemble methods (AdaBoost, XGBoost, Random Forest, etc.) that use multiple neural networks with different architectures. However, this requires training and storing multiple models, which becomes a major challenge as the network gets deeper.

So there's a great solution called Dropout Layers.

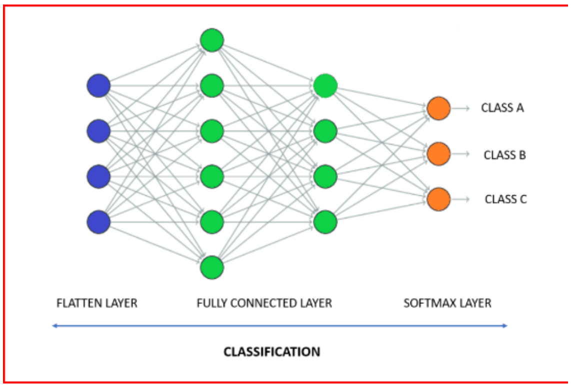


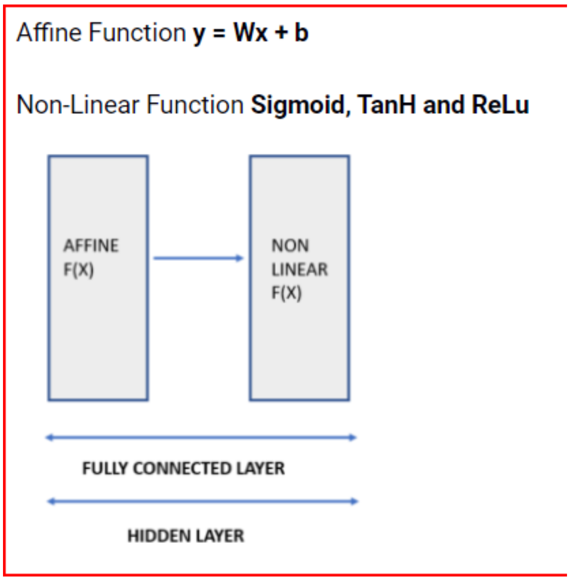
(a) Standard Neural Net (b) After applying dropout.

3.4.4 Fully Connected Layer:

Convolutional Neural Networks (CNNs), which have been demonstrated to be particularly useful in detecting and classifying pictures for computer vision, must include fully linked layers. Convolution and pooling, which divide the image into features and analyse each one separately, are the first steps in the CNN process. A fully connected neural network structure receives the output of this procedure and uses it to determine the final classification.

The final layer of the convolutional neural network is the Fully Connected Layer, sometimes referred to as the Hidden Layer. Affine and non-linear functions are combined in this layer.





Flatten Layer, a one-dimensional layer, provides input to the fully connected layer (1D Layer). The Affine function receives the data from the Flatten Layer first, followed by the Non-Linear function. One FC (Fully Connected) or one Hidden Layer is the combination of one Affine function and one Non-Linear function.

Depending on how deep we want to go with our categorization model, we can add a number of these layers. Be aware that the training dataset is solely responsible for this. In order to determine the probability distribution over the final set of all classes, the output from the last hidden layer is supplied to the Softmax or Sigmoid function.

The Deep Neural Network's Classification section includes the Flatten Layer, Fully Connected Layer, and Softmax Layer combinations.

Looking at the entire neural network, we can observe that the first layers of the convolutional neural network are made up of:

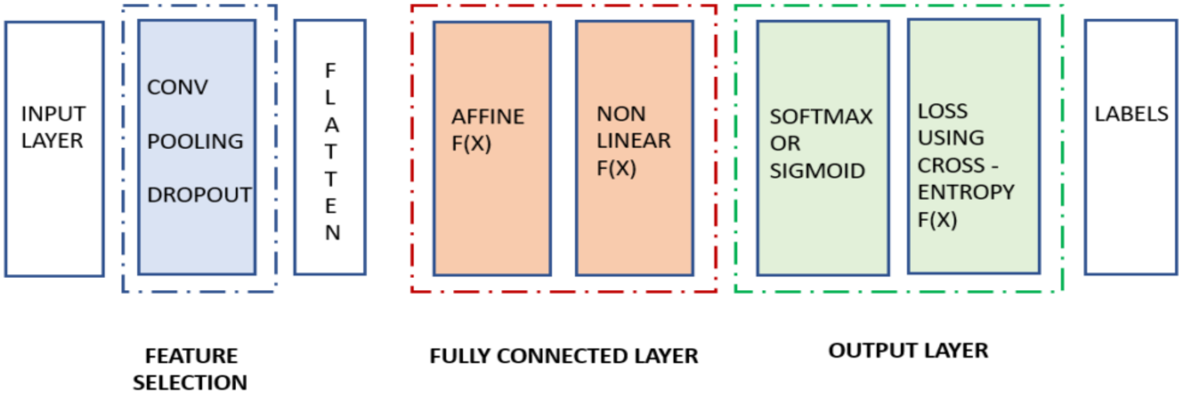
Pooling Layer Dropout Layer Convolutional Layer

These three together make up feature choice (extraction). One can add numerous permutations and combinations of these layers based on the training data.

The convolutional neural network's output layer consists of:

Calculating Softmax or Sigmoid Layer Losses Using the Cross-entropy Function

The list of all classes, for instance 10 classes, along with the probabilities assigned to each class, constitutes the final calculation of classes (Labels). The final class of the input image is the one with the highest probability.



3.4.5 SOFT MAX LAYER:

A softmax layer applies a softmax function to the input.

A generalisation of the logistic function, the softmax activation function or normalised exponential function transforms a vector of K real values into a vector of K real values that add to 1. The softmax function turns every number between 0 and 1 regardless of whether the input values are negative, zero, positive, or more than 1. To understand them as probabilities, this is done.
The softmax function converts any input that is negative or little in value into a small probability. In contrast, it converts a significant input value into a large likelihood. However, the values will always range from 0 to 1.

Full softmax and candidate sampling are two variations of the softmax function.

i. Full softmax :

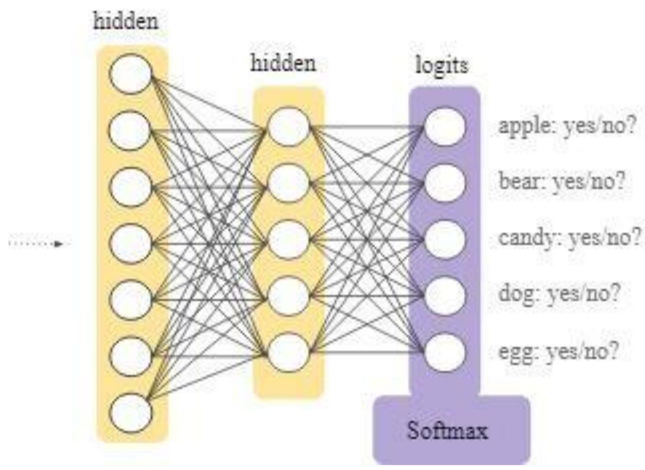
   This form of softmax determines the probabilities for each potential class. It will be especially useful when working with Python's multiclass neural networks. When only a few classes use it, it is relatively affordable. But as soon as the number of classes rises, it gets pricey.

ii. Candidate sampling:

   Only the probability of labels that are positive is calculated in this version of the softmax function. It only does this, though, for a selection of unfavourable labels. With this variation, the theory is that the negative classes can benefit from the less common negative reinforcement.

   Candidate sampling is acceptable as long as the classes that perform well receive enough encouragement. Obviously, in order to guarantee computational effectiveness, this needs to be observed empirically. However, when there are more classes to take care of, it increases the output's efficiency overall.

For instance, we don't need to supply the probabilities for a non-fruit example if our goal is to identify if the input image is an apple or a mango.



Properties of Softmax layer:

Layer name, given as a string scalar or a character vector. The trainNetwork, assembleNetwork, layerGraph, and dlnetwork methods automatically give names to layers with the name "" for Layer array input.

Input of softmax layer:

This property can only be read.
number of the layer's inputs. This layer only accepts one input.

Types of Data: double

Output of softmax layer:

This property can only be read.
number of the layer's outputs. This layer gives only one output.

Types of Data: double

3.4.6 Classification Layer:

For classification and weighted classification problems with classes that are mutually exclusive, a classification layer calculates the cross-entropy loss.

From the output size of the preceding layer, the layer infers the number of classes. Before the classification layer, for instance, you may include a fully connected layer with an output size of K and a softmax layer to specify the network's K number of classes.

Classification Output:

A vector of positive values, "none," or the class weights for the weighted cross-entropy loss.

Each element of a vector class weight corresponds to a class's weight in the Classes property. You must also define the classes using 'Classes' in order to specify a vector of class weights.

Unweighted cross-entropy loss is used by the layer if the ClassWeights attribute is set to "none".

Sorts of output layer classes:

Categorical vector, string array, cell array of character vectors, or "auto" are the possible output layer classes. When training time comes around, the software automatically adjusts the classes if Classes is set to "auto". The software changes the output layer's classes to categorical if you specify a string array or a cell array of character vectors (str,str).

Data Types: Cell, String, Categorical, and Char

Output size:

This property can only be read-only.

Integer positive value indicating the output's size. This number represents how many labels there are in the data. The output size is predetermined to be "auto" before training.

Number of Inputs:

This property can only be read.

number of the layer's inputs. This layer only accepts one input.

Types of Data: double

Input Names:

This property can only be read.

Enter the layer names here. This layer only accepts one input.

data type:cell

Number of outputs:
The layer's total number of outputs. It has no outputs.

Types of Data: double

Output Names:

Publish the layer names. It has no outputs.

Cell data types

4.Code Explanation:

Audioread():

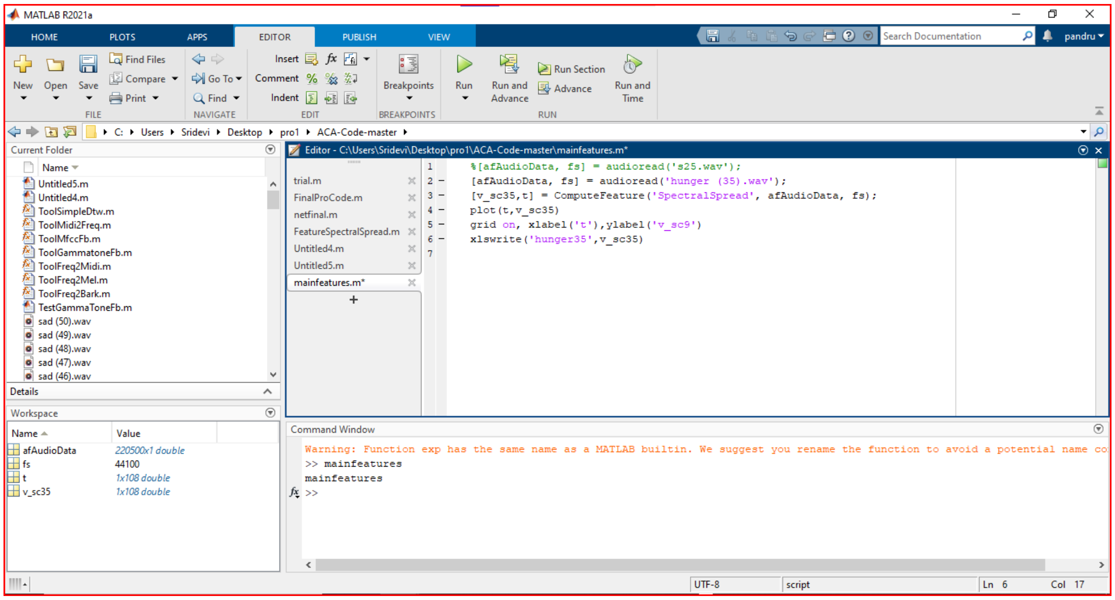
Get information about the audio file, write data to it, and then read the data back into the MATLAB® workspace.
The workspace now has an audio data matrix (y) and a sampling rate (Fs).
To save the information to a WAVE file with the name handel.wav in the current folder, use the audio write function.
Other audio file types can be written to using the audio write function. See Supported File Formats for Import and Export for a complete list of supported formats.

ComputeFeature():

Many computer vision algorithms' building pieces are local features and their descriptions. Image registration, object detection and categorization, tracking, motion prediction, and content-based image retrieval are some of their uses (CBIR). To more effectively manage scale changes, rotation, and occlusion, these algorithms make advantage of local features. The corner detectors FAST, Harris, and Shi & Tomasi, as well as the blob detectors SIFT, SURF, KAZE, and MSER, are all part of the Computer Vision Toolbox TM. The descriptors SIFT, SURF, FREAK, BRISK, LBP, ORB, and HOG are part of the toolset. Depending on the needs of your application, you can combine and match the detectors and descriptors.

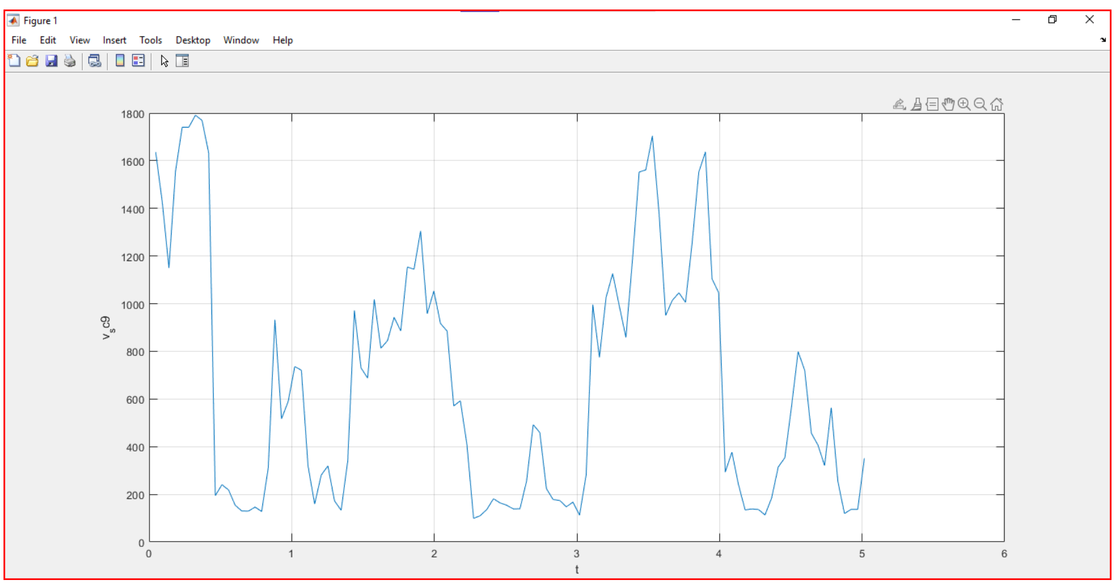
Xlsread():

The xlsread function returns the second output from processFcn in array custom, the numeric and text data from cell array raw, and the text fields from cell array txt. The data that is stored in the spreadsheet is not altered by the xlsread function. Only Windows machines running Excel can use this syntax.



Feature extraction code

We gave the collected audio in .wav from to get it read using audio read function then we computed our desired feature which is spectral spread then we stored the feature extracted into the xls sheet using xls write function.we can plot the results for our observation.



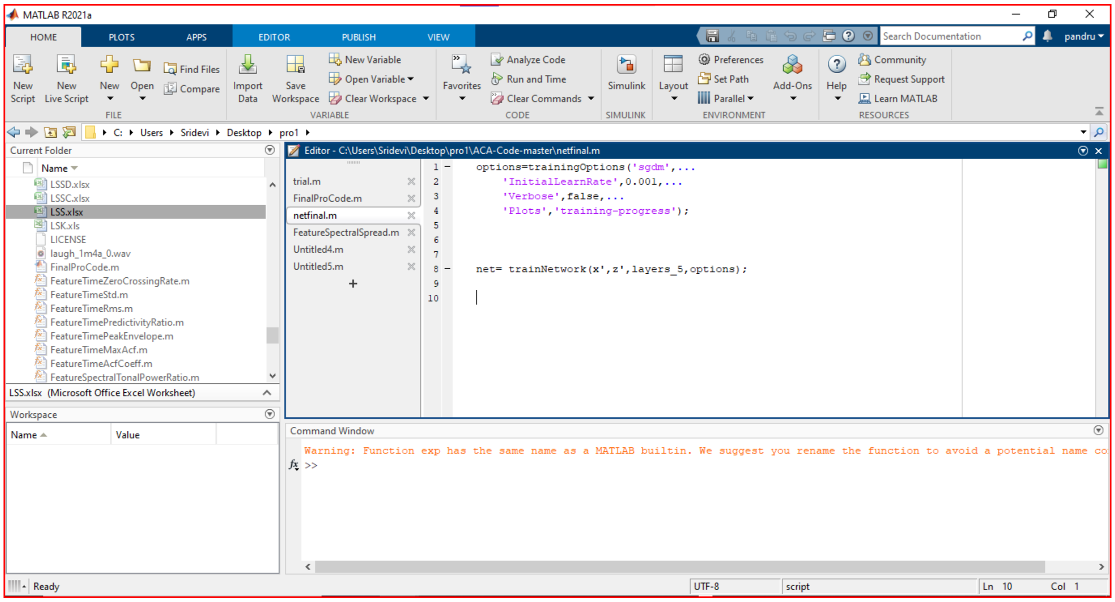
Feature observation

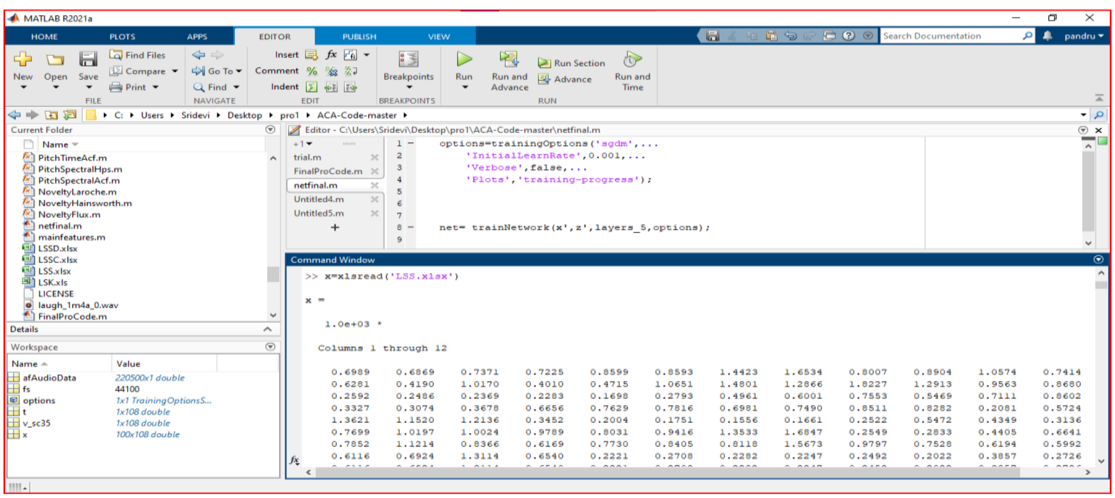
The stochastic gradient descent with momentum (SGDM) optimizer:

Each iteration is an estimation of the gradient and an update of the network parameters. The gradient of any line or curve tells us the rate of change of one variable with respect to another.

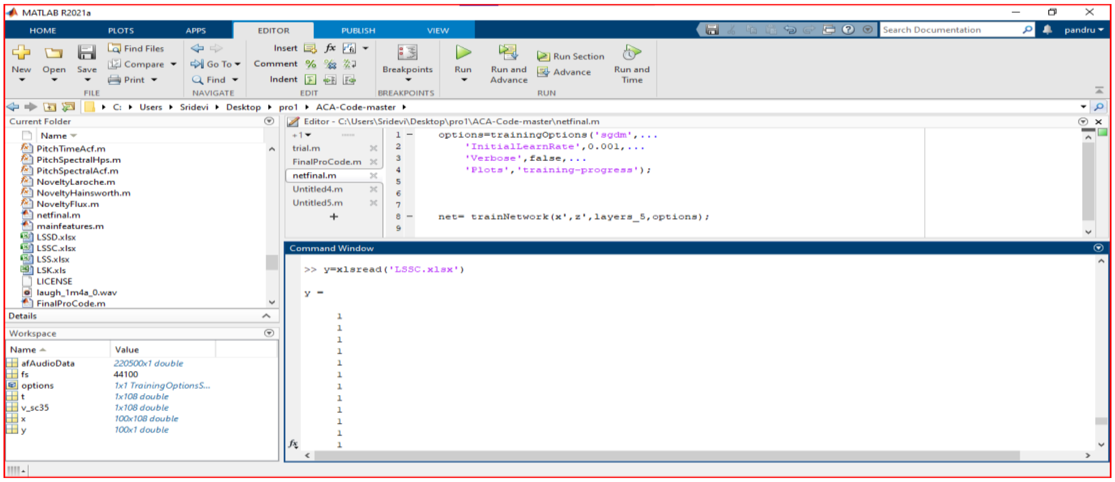
Verbose:

verbose is the choice that how you want to see the output of your Neural Network while it's training. If you set verbose = 0, It will show nothing.

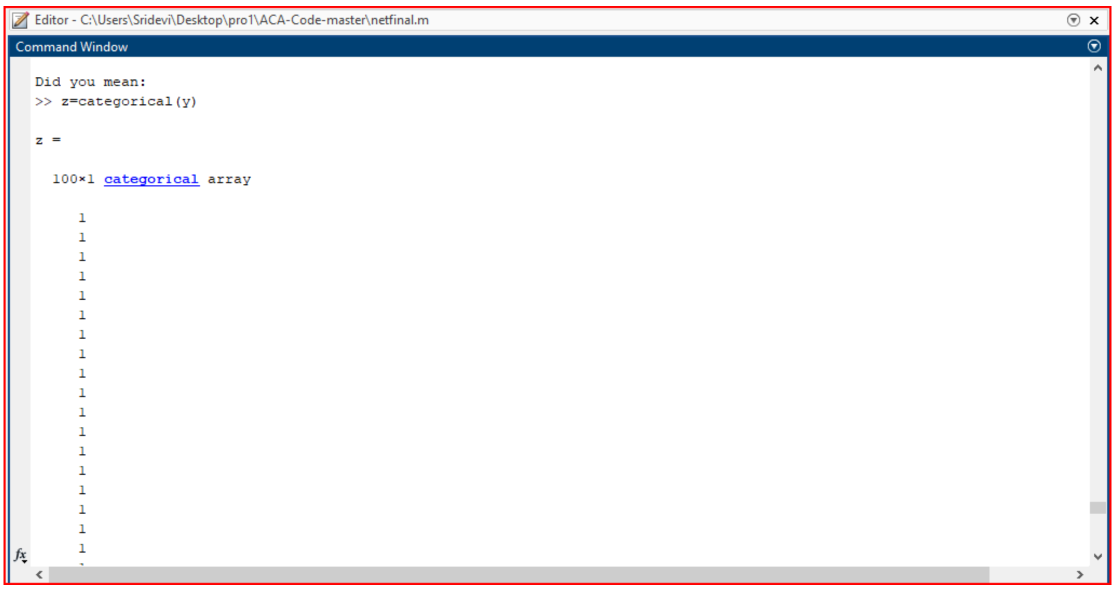




Assigning variable x to the LSS i.e; file of extracted features.



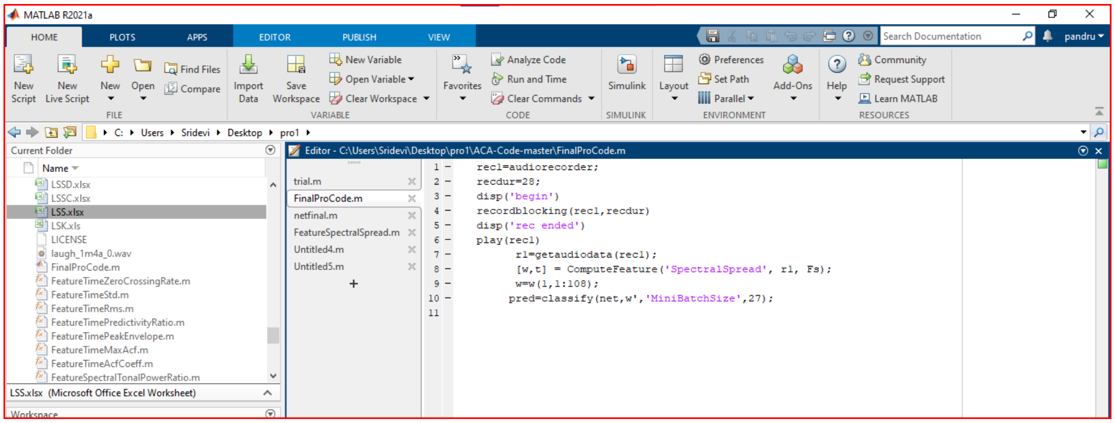
Assigning classifications of the features to the variable y.



Categorical():

Data with values from a limited number of discrete categories are stored as categorical data types. For instance, the categorical array C = categorical('R','G','B','B','G','B') contains six elements that can be either from the categories R, G, or B.

A categorical array maintains meaningful labels for the data values while also enabling effective storage and simple manipulation of non-numerical data. The categories can be arranged naturally, although it's not necessary.



Currently, we are utilizing the audio recorder in Matlab to record audio input using a microphone; setting the recording duration is necessary; we set it to 28 seconds, and then we began recording using recordblocking ()

After computing feature extraction, we extracted data from the recorded audio, fed it into the network, and categorised the results.

As hunger and pain are out two main emotional concerns of infants , we gave the hunger set a 1 and the pain set a 0.

Conclusion/Results:

```
pred =

  categorical

     1
```

As 1 is predicted this shows baby is hungry

Future Work:

Since we already examined two of the infants's emotions, we want to continue by examining other emotions, such as fear and specific causes of the infant's crying. To be more specific, we considered adding a few other elements, such as spectralrolloff,s pectralkurtosis etc..

References:

- H. Karp, The Happiest Baby on the Block; Fully Revised and Updated Second Edition: The New Way to Calm Crying, New York City, NY, USA, 2015.

- J. A. Green, P. G. Whitney, and M. Potegalb, “Screaming, yelling, whining and crying: categorical and intensity differences in vocal expressions of anger and sadness in children’s tantrums,” Emotion, vol. 5, no. 11, pp. 1124−1133, Oct. 2011.

- Y. Kheddache and C. Tadj, “Acoustic measures of the cry characteris tics of healthy newborns and newborns with pathologies,” Journal of Biomedical Science and Engineering, vol. 6, no. 8, 9 pages, 2013.

- L. Liu, K. Kuo, and Sen M. Kuo, “Infant cry classification integrated ANC system for infant incubators,” in Proc. IEEE International Conf. on Networking, Sensing and Control, Paris, France, 2013, pp. 383−387.

- L. Liu and K. Kuo, “Active noise control systems integrated with infant cry detection and classification for infant incubators,” in Proc. Acoustic, pp. 1−6. 2012.

- L. LaGasse, A. Neal, and M. Lester, “Assessment of infant cry: acoustic cry analysis and parental perception,” Ment Retard Dev Disabil Res Rev., vol. 11, no. 1, pp. 83−93, 2005.

- Varallyay Jr. Gy ´ orgy, “Future prospects of the application of the infant ¨ cry in the medicine,” Periodica Polytechnica Ser. El. Eng, vol. 50, no. 1−2, pp. 47−62, 2006.

- G. Buonocore and C.V. Bellieni, Neonatal Pain, Suffering, Pain and Risk of Brain Damage in the Fetus and Newborn, Berlin, Germany, Springer, 2008.

- L. L. LaGasse, R. Neal, and B. M. Lester. “Assessment of infant cry: acoustic cry analysis and parental perception,” Mental Retardation and Developmental Disabilities Research Reviews, vol. 11, no. 1. pp. 83−93, 2005.

- L. Tan and J. Jiang, Digital Signal Processing: Fundamentals and Applications (3rd edition). Cambridge, MA, USA, Academic Press, 2017.

- Z. Ren, K. Qian, Z. X. Zhang, V. Pandit, A. Baird, and B. Schuller

- Dong Yu and Jinyu Li. “Recent progresses in deep learning based acoustic models,” IEEE/CAA J. Autom. Sinica, vol. 4, no. 3, pp. 396−409, April 2017

- B. Goldand N. Morgan, Speech and Audio Signal Processing. New York, NY, USA, John Wiley & Sons, 2011.

- V. R. Fisichelli, S. Karelitz, C. F. Z. Boukydis, and B. M. Lester, “The cry attencies of normal infants and those with brain damage,” Infant Crying, Plenum

Press, 1985.

- C. F. Z. Boukydis and B. M. Lester, Infant Crying: Theoretical and Research Perspectives, Berlin, Germany, Springer Science and Bussiness Media, 2012.

- S. Ludington-Hoe, X. Cong, and F. Hashemi, “Infant crying: nature, physiologic consequences, and select interventions,” Neonatal Netw. vol. 21, no. 2, pp. 29−36. Mar. 2002.

- P. Dunstan, Calm the Crying: The Secret Baby Language That Reveals the Hidden Meaning Behind an Infant’s Cry, New York City, NY, USA, Avery, 2012.

- M. Sahidullah, and G. K. Saha, “Design analysis and experimental eval uation of block based transformation in MFCC computation for speaker recognition,” Speech Communication, vol. 54, no. 4, pp. 543−565, May 2012.

- F. Katzberg, R. Mazur, M. Maass, P. Koch, and A. Mertins, “A compressed sensing framework for dynamic sound-field measurements,” IEEE/ACM Trans. Audio, Speech, and Language Processing, vol. 26, no. 11, pp. 1962−1975, Jun. 2018.

- D. Needell and R. Ward, “Two-subspace projection method for coherent overdetermined systems,” Journal of Fourier Analysis and Applications, vol. 19, no. 2, pp. 256−269, April, 2013.

- C. Lau, “Development of suck and swallow mechanisms in infants,” Ann. Nutr. Metab., vol. 7, no. 5, pp. 7−14, July 2015.

- P. Runefors and E. Arnbjonsson, “A sound spectrogram analysis of ¨ children’s crying after painful stimuli during the first year of life,” Folia honiatr. Logop., vol. 2, no. 57, pp. 90−95, Mar–Apr. 2005.

- Lichuan Liu, Senior Member, IEEE, Wei Li, Senior Member, IEEE, Xianwen Wu, Member, IEEE and Benjamin X. Zhou.

Plagiarism Report

CHILD CRY P

ORIGINALITY REPORT

16%

SIMILARITY INDEX

PRIMARY SOURCES

1	resources.system-analysis.cadence.com Internet	160 words — 3%
2	docs.nvidia.com Internet	139 words — 2%
3	Lawrence R. Rabiner, Ronald W. Schafer. "Introduction to Digital Speech Processing", Foundations and Trends® in Signal Processing, 2007 Crossref	91 words — 2%
4	www.ijera.com Internet	74 words — 1%

5	en.wikipedia.org Internet	73 words — 1%
6	er.ucu.edu.ua Internet	44 words — 1%
7	Sana Khanam, Safdar Tanweer, Syed Khalid. "Artificial Intelligence Surpassing Human Intelligence: Factual or Hoax", The Computer Journal, 2020 Crossref	40 words — 1%
8	sjii.indexedresearch.org Internet	31 words — 1%
9	Lopes, Paulo Alexandre da Silva. "Program and Aspect Metrics for MATLAB: Design and Implementation", Universidade do Minho (Portugal), 2021 ProQuest	30 words — 1%

10	Jūratė Vaičiulytė, Leonidas Sakalauskas. "Recursive estimation of multivariate hidden Markov model parameters", Computational Statistics, 2019 Crossref	28 words — < 1%
11	ijesrt.com Internet	27 words — < 1%
12	Yijing Zhao, Zheng Zheng, Yang Liu. "Survey on Computational-Intelligence-Based UAV Path Planning", Knowledge-Based Systems, 2018 Crossref	23 words — < 1%
13	ruor.uottawa.ca Internet	22 words — < 1%
14	eprints.utm.my Internet	20 words — < 1%
15	Bengis, Merrick Kenna. "Digital Environment Evolution Modelling and Simulation", University of Johannesburg (South Africa), 2021 ProQuest	17 words — < 1%
16	Lukáš Rapant, Kateřina Slaninová, Jan Martinovič, Tomáš Martinovič. "Chapter 15 Traffic Speed Prediction Using Hidden Markov Models for Czech Republic Highways", Springer Nature, 2016 Crossref	16 words — < 1%
17	umpir.ump.edu.my Internet	15 words — < 1%
18	N.Uma maheshwari*, Dr. Archek Praveen Kumar, K. Narmada, Affrose, B. Sneha. "Continuous Telugu Speech Recognition on T-LPC and DNN Techniques", International Journal of Recent Technology and Engineering (IJRTE), 2019 Crossref	14 words — < 1%
19	mindmajix.com Internet	14 words — < 1%

20	gradesfixer.com Internet	11 words — < 1%
21	www.theijes.com Internet	11 words — < 1%
22	dalpero.it Internet	10 words — < 1%
23	Darwish, Rami Hilmi. "Enhancing the Ontologies Matching in Semantic Web Using Artificial Neural Network", Princess Sumaya University for Technology (Jordan), 2021 ProQuest	9 words — < 1%
24	coek.info Internet	9 words — < 1%
25	"Introduction: Human and Computational Mind", Studies in Computational Intelligence, 2007 Crossref	8 words — < 1%
26	www.ijser.org Internet	8 words — < 1%