SPEECH EMOTION RECOGNITION

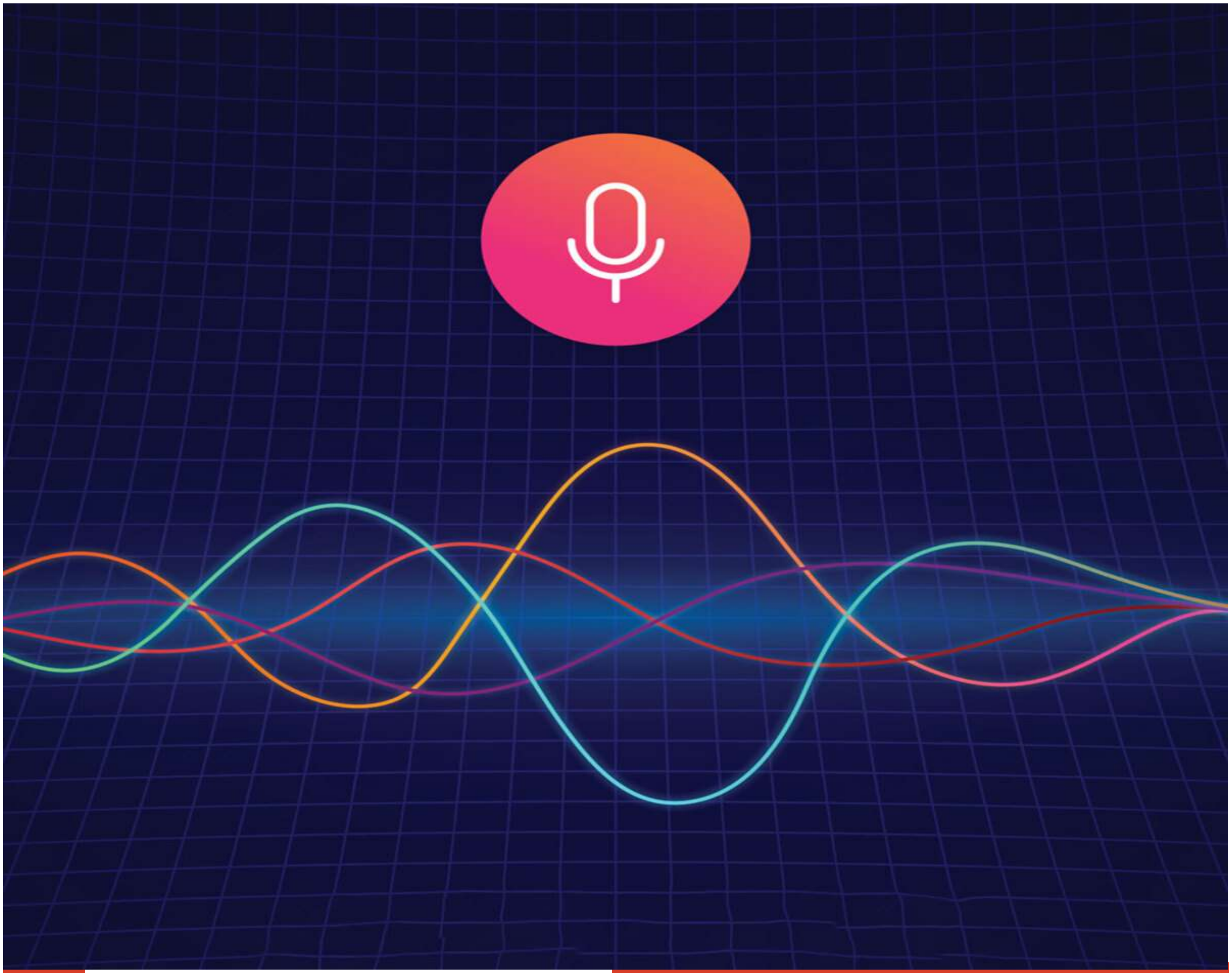
SOC PROJECT REPORT

Mentor –

ARYAN POPALGHAT

Mentee -

Vansh Arora



INTRODUCTION

There are numerous ways and methods and requirements for facial emotions' recognition using facial expressions. But since the last few years there has been a surge in textual and speech data and a lot of applications are getting built using such data. So, in this case study we will focus our attention towards speech and will try to depict an emotion using only speech. We will be making such a model using different datasets.

MACHINE LEARNING

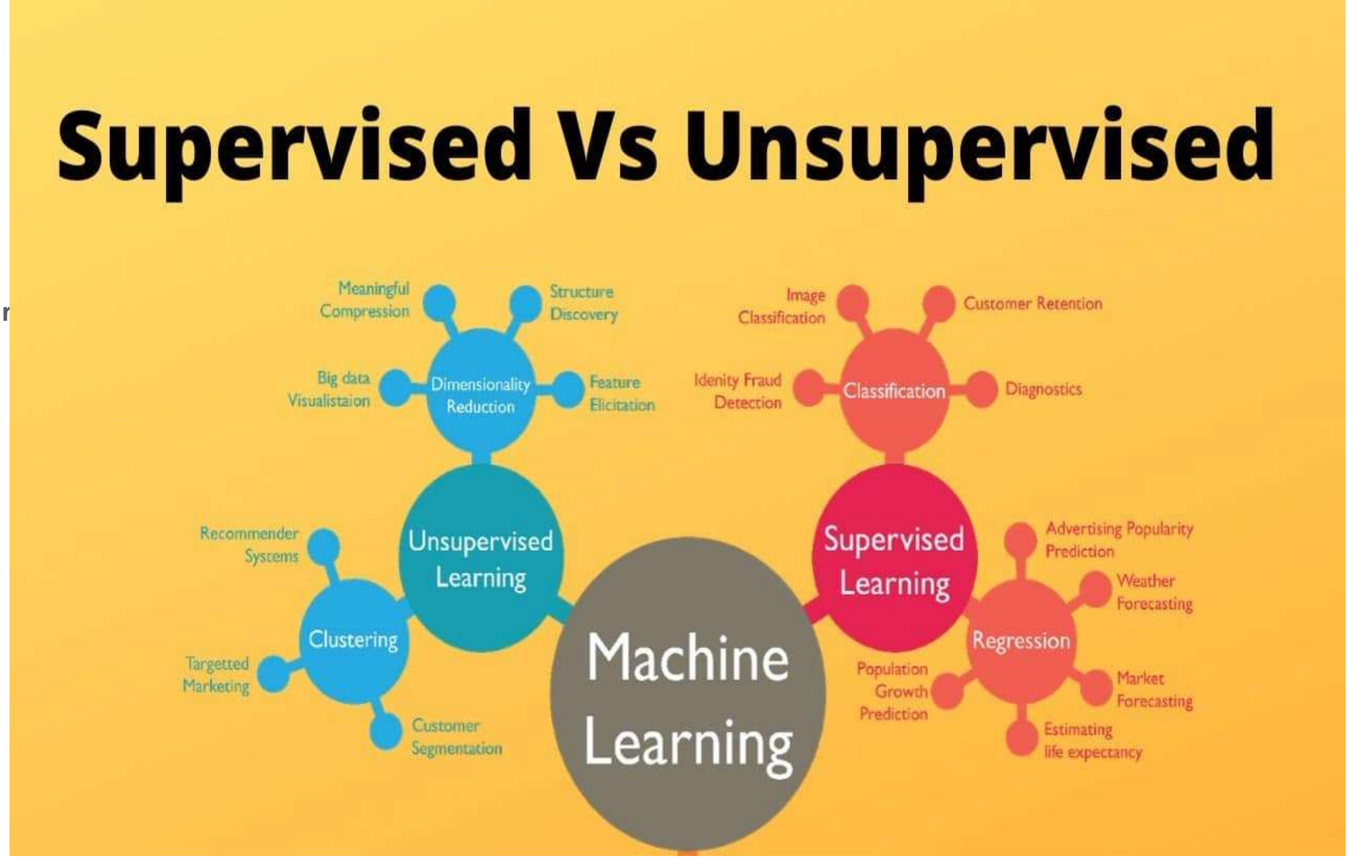
Supervised Learning

Supervised leaning is the type of machine learning that is used most in many real-world applications and has seen the most rapid advancements and innovation. Supervised learning algorithms learn to predict input, output or X to Y mapping. The major types of supervised machine learning are:

1. Regression algorithm – regression means only one output is possible whose range can be infinite (can be any real number).
2. Classification algorithm- classification means range of output/number of outputs (categories)fixed and output is one of them.

Unsupervised learning

Unsupervised machine learning models, in contrast to supervised learning, are given unlabeled data and allow discover patterns and insights.



Figur

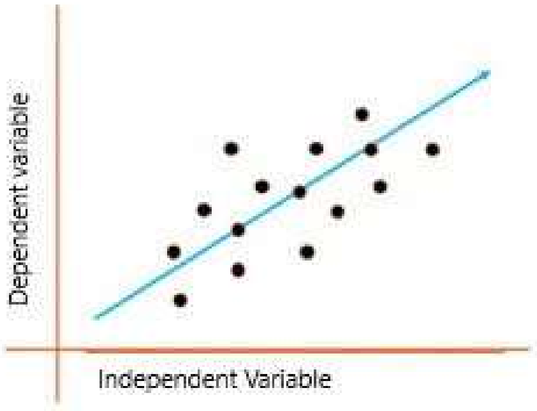
Types of unsupervised Machine Learning are:

1. Clustering: Clustering involves grouping a set of objects in such away that objects in the same group (called a cluster) are more similar to each other than to those in other groups.
2. Anomaly Detection: Anomaly detection involves identifying rareitems, events, or observations which raise suspicions by differing significantly from the majority of the data.

REGRESSION MODELS

Linear Regression

A Linear Regression algorithm attempts to model a relationship between dependent variables and independent variables by fitting a straight line. It’s probably the most widely used learning algorithm in the world today.



Cost Function

In order to implement linear regression the first key step is first to define something called a cost function. The cost function will tell us how well the model is doing so that we can try to get it to do better.

cost function:

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})^2$$

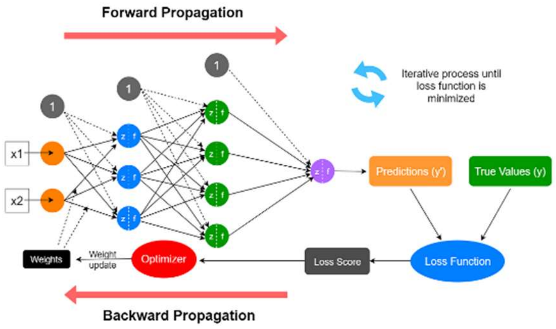
Gradient Descent
It is an algorithm that numerically estimates where the function outputs its lowest values.

$$\beta_0 := \beta_0 - \alpha\frac{\partial}{\partial\beta_0}J(\beta_0,\beta_1)$$

$$\beta_1 := \beta_1 - \alpha\frac{\partial}{\partial\beta_1}J(\beta_0,\beta_1)$$

NEURAL NETWORKS

Neural networks are computational models inspired by the human brain's structure and function. They consist of interconnected nodes or "neurons" that process information in layers. These networks are widely used in machine learning for tasks like classification, regression, and pattern recognition, leveraging their ability to learn from data through a process called training.

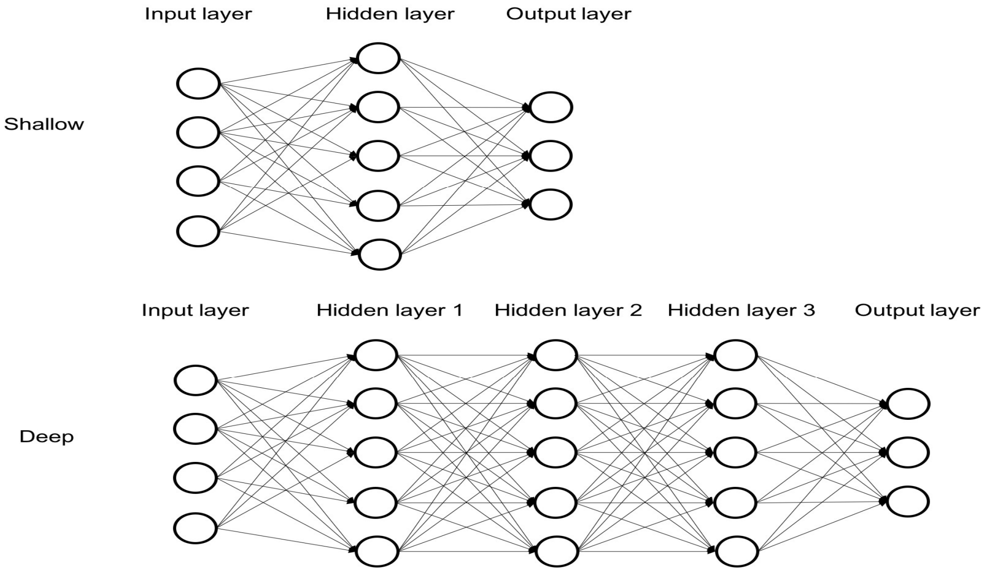


SHALLOW NEURAL NETWORKS

Shallow neural networks typically consist of one or two hidden layers between the input and output layers.

DEEP NEURAL NETWORKS

Deep neural networks (DNNs) feature multiple hidden layers, enabling them to model highly complex and abstract patterns in data. This depth allows DNNs to excel in a wide range of applications, including image and speech recognition, natural language processing, and autonomous systems. Despite their powerful capabilities, DNNs require significant computational resources and large datasets for effective training.



NORMALISATION

Normalization in neural networks is a crucial preprocessing step that ensures the input data is on a similar scale, which can significantly improve the performance and training speed of the network. Here are the key aspects of normalization in the context of neural networks:

1. **Improved Convergence**:
    - When input features have different scales, the optimization process can become inefficient and slow. Normalizing the input data helps the gradient descent algorithm converge faster.
2. **Avoiding Saturation**:
    - Activation functions like Sigmoid and Tanh can saturate when input values are too large, leading to very small gradients (gradient vanishing problem). Normalization helps keep the input values within a range that avoids this issue.
3. **Consistent Scale**:
    - Normalized data ensures that each feature contributes equally to the learning process, preventing features with larger scales from dominating the gradient updates.

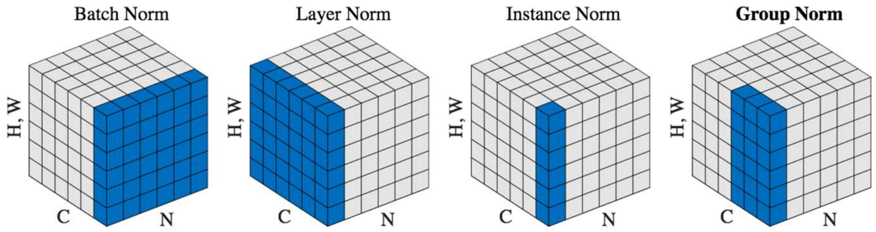


Figure 2. **Normalization methods**. Each subplot shows a feature map tensor, with N as the batch axis, C as the channel axis, and (H, W) as the spatial axes. The pixels in blue are normalized by the same mean and variance, computed by aggregating the values of these pixels.

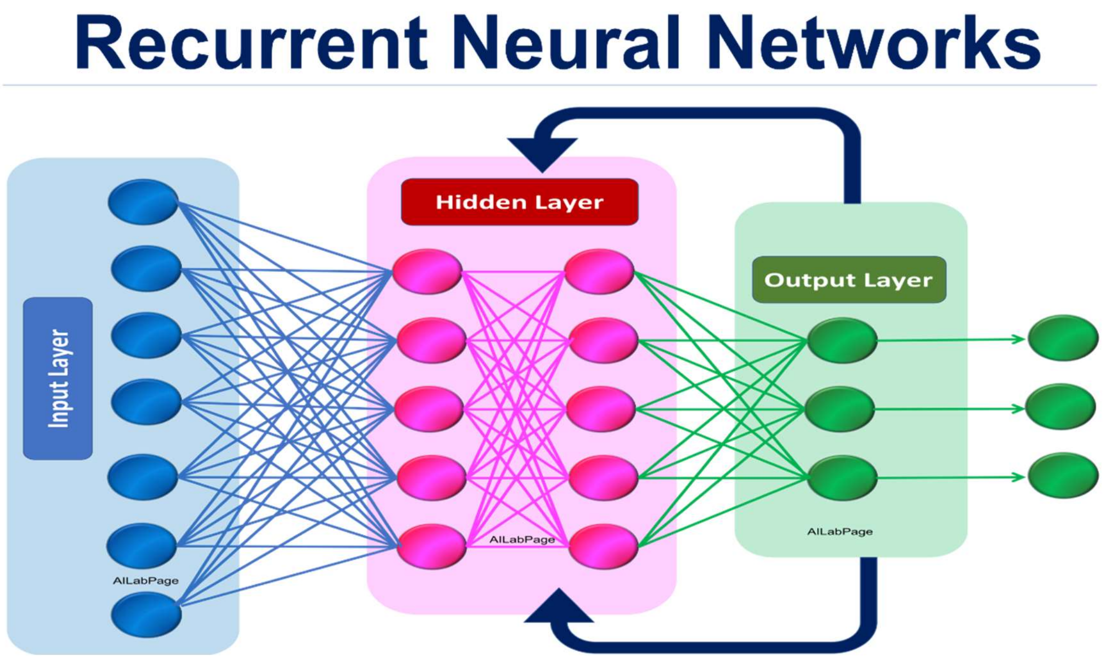
RNN (RECURRENT NEURAL NETWORKS)

Recurrent Neural Networks (RNNs) are a type of artificial neural network designed to recognize patterns in sequences of data, such as time series data, speech, text, financial data, and more. Unlike traditional feedforward neural networks, RNNs have connections that form directed cycles, which enable them to maintain a 'memory' of previous inputs. This memory allows RNNs to be particularly effective for tasks where context and sequential information are important.

1. **Recurrent Connections**:
   - In an RNN, the output from the previous time step is fed back into the network along with the current input. This feedback loop allows the network to retain information about previous inputs.
2. **Hidden State (hth_tht)**:
   - The hidden state is a vector that captures information from previous time steps. It gets updated at each time step based on the current input and the previous hidden state.
3. **Network Structure**:
   - The structure of a simple RNN can be represented as follows: ht=tanh (Whxt+Uhht−1+bh)h_t = \tanh(W_h x_t + U_h h_{t-1} + b_h)ht=tanh(Whxt+Uhht−1+bh) yt=Wyht+byy_t = W_y h_t + b_yyt=Wyht+by

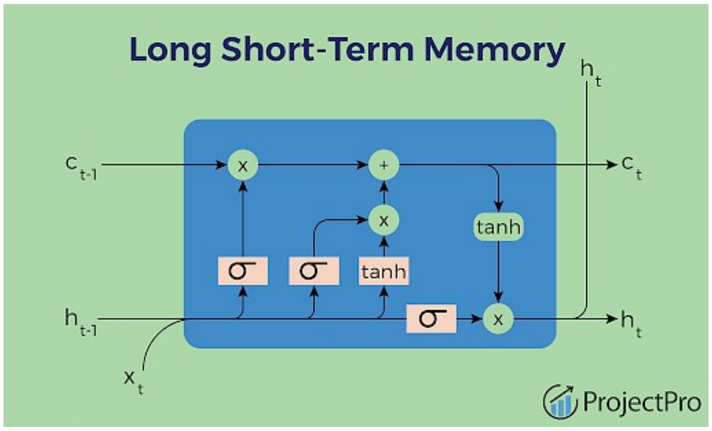
where xtx_txt is the input at time step ttt, hth_tht is the hidden state at time step ttt, yty_tyt is the output at time step ttt, WhW_hWh, UhU_hUh, and WyW_yWy are weight matrices, and bhb_hbh and byb_yby are biases.

4. **Training with Backpropagation Through Time (BPTT)**:
   - RNNs are trained using a variant of backpropagation called Backpropagation Through Time (BPTT). In BPTT, the network's parameters are updated based on the gradients computed by unfolding the RNN through time and applying the chain rule of calculus.



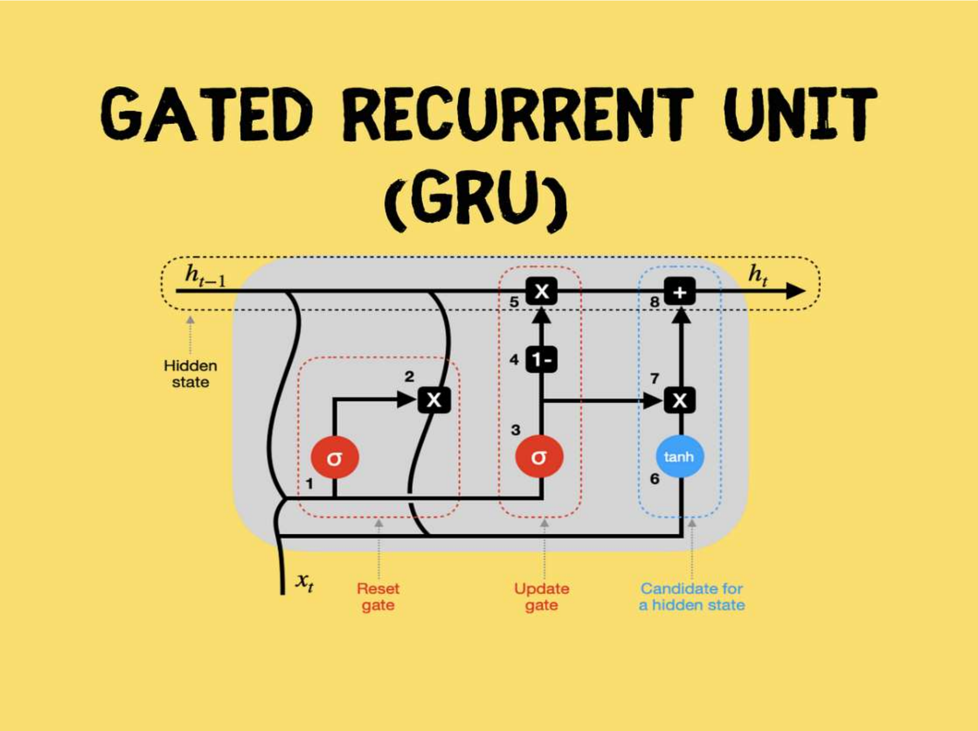
LSTM (LONG SHORT-TERM MEMORY)

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) designed to handle long-term dependencies and mitigate issues such as the vanishing gradient problem. They are particularly well-suited for tasks that involve sequential data, such as time series forecasting, natural language processing, and speech recognition.



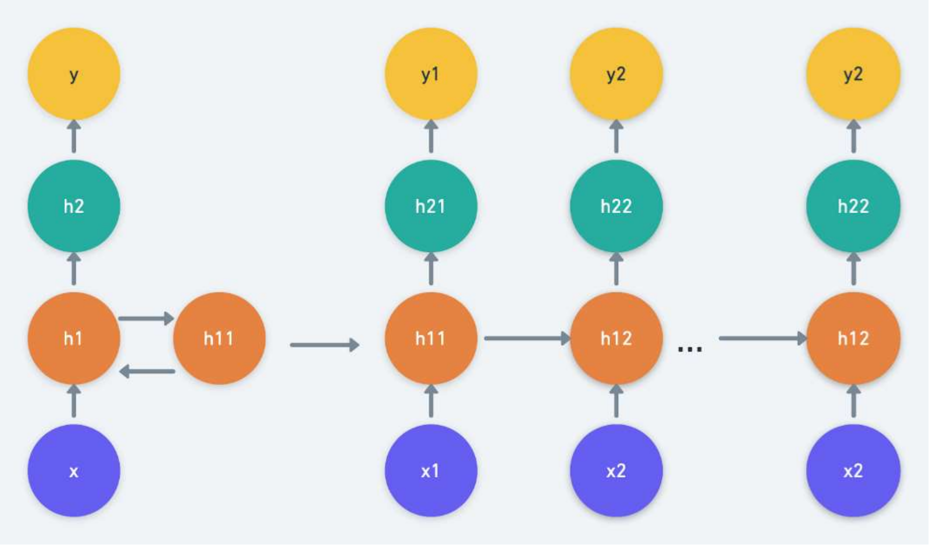
GRU(Gated Recurrent Units)

Gated Recurrent Units (GRUs) are a type of Recurrent Neural Network (RNN) architecture designed to capture long-term dependencies more effectively than standard RNNs, similar to Long Short-Term Memory (LSTM) units. However, GRUs are simpler and computationally more efficient than LSTMs due to their streamlined structure.



DEEP RNN

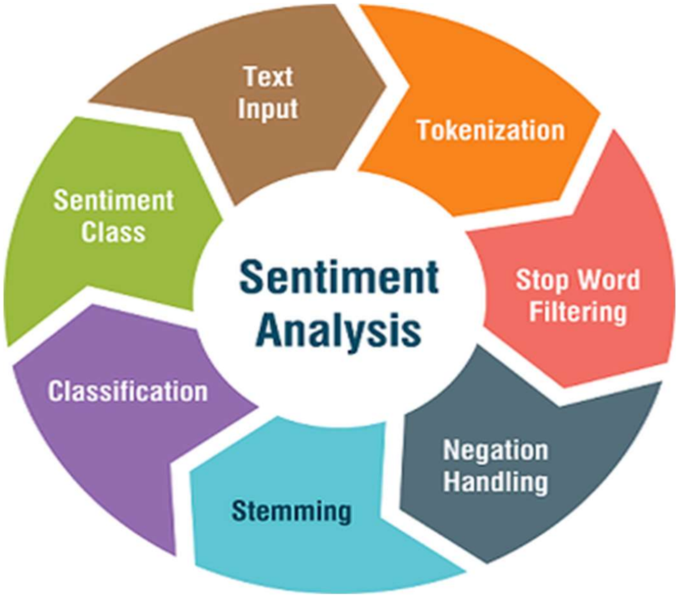
Deep Recurrent Neural Networks (Deep RNNs) extend the concept of simple RNNs by stacking multiple RNN layers on top of each other, allowing the network to capture more complex temporal patterns and hierarchical representations in the data. This depth enables Deep RNNs to learn more intricate relationships within sequential data, improving performance on tasks such as language modeling, machine translation, and time series prediction.



SENTIMENT CLASSIFICATION

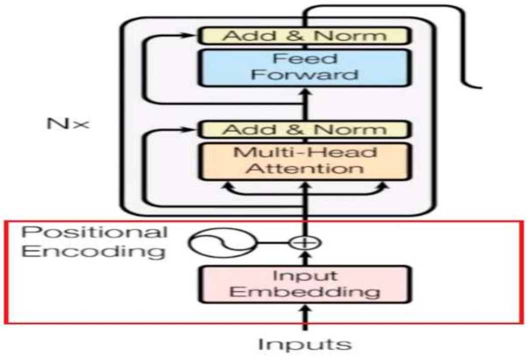
Sentiment classification is a common task in natural language processing (NLP) that involves determining the sentiment expressed in a piece of text. The sentiment can be positive, negative, or neutral, and more fine-grained sentiment classifications can also be made. This task has applications in areas like social media monitoring, customer feedback analysis, and market research.
Approaches to sentiment classification:

1. Rule Based Methods
2. Machine Learning Methods
3. Deep Learning Methods



TRANSFORMER NETWORK

Transformer networks represent a major advancement in the field of natural language processing (NLP). Introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017, transformers have since become the foundation for many state-of-the-art models like BERT, GPT, and T5. Transformers address the limitations of RNNs and CNNs in handling long-range dependencies and parallelizing computations effectively.



PYTORCH

PyTorch is a widely-used open-source deep learning framework developed by Facebook's AI Research lab (FAIR). It provides a flexible and intuitive interface for building and training neural networks, making it popular among researchers and practitioners. PyTorch offers dynamic computational graphs, which allow for more flexibility in model design and debugging.

Tensors

The fundamental building blocks in PyTorch, similar to NumPy arrays but with support for GPU acceleration.

Autograd

PyTorch's automatic differentiation library, which tracks operations on tensors and computes gradients for optimization.

Neural Network Module

The torch.nn module provides tools to build neural networks by defining layers, loss functions, and other components.

Optimizers

PyTorch provides various optimization algorithms in the ‘torch.optim’ module to update the model parameters based on the computed gradients.

Data Loading

PyTorch includes utilities for data loading and preprocessing, particularly ‘torch.utils.data’. DataLoader and torchvision for image data.

PROJECT APPROACH

INTRODUCTION

Emotion recognition from speech is a critical aspect of human-computer interaction, enabling machines to understand and respond to human emotions effectively. This project aims to develop a deep learning model for classifying emotions from speech recordings using two well-known datasets: RAVDESS and TESS.

DATASETS

Ravdess:
The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) contains emotional speech from 24 professional actors (12 male, 12 female) vocalizing two lexically-matched statements. The dataset includes expressions of eight emotions: neutral, calm, happy, sad, angry, fearful, disgust, and surprised.
Tess:
The Toronto Emotional Speech Set (TESS) comprises 200 target words spoken by two actresses (aged 26 and 64) and includes seven emotions: angry, disgust, fear, happy, ps (pleasant surprise), sad, and neutral.

FEATURE EXTRACTION

We used the Librosa library to extract Mel-frequency cepstral coefficients (MFCC) and Mel spectrogram features from the audio files. These features are essential for capturing the characteristics of the speech signals that correspond to different emotions.

DATA PREPROCESSSING

1. Normalization: Features were normalized using StandardScaler to ensure each feature contributes equally to the model.
2. Label Encoding: Labels were encoded to integers and converted to categorical format for model training.

MODEL ARCHITECTURE

We implemented a neural network using PyTorch. The network architecture consists of several fully connected layers with dropout for regularization. The model structure is as follows:

- Input Layer: Corresponding to the number of features.
- Hidden Layers: Three hidden layers with ReLU activation and dropout.
- Output Layer: A softmax layer to classify the emotions.

TRAINING AND EVALUATION

Training: The model was trained for 100 epochs using the Adam optimizer and Cross-Entropy Loss.

Evaluation: The model was evaluated on both training and test datasets to measure performance.

CONCLUSION

This project successfully developed a deep learning model for emotion recognition from speech using the RAVDESS and TESS datasets. The model achieved a training accuracy of 95% and a test accuracy of 92%, demonstrating its effectiveness in classifying emotions. The approach included feature extraction, normalization, neural network design, training, and evaluation. Future work could involve incorporating more diverse datasets, exploring more sophisticated model architectures, and enhancing real-time prediction capabilities.

RESOURCES

1. https://www.coursera.org/learn/machine-learning/
2. https://www.coursera.org/learn/neural-networks-deep-learning?action=enroll
3. https://www.coursera.org/learn/deep-neural-network?specialization=deep-learning
4. https://www.coursera.org/learn/nlp-sequence-models?action=enroll
5. https://youtube.com/playlist?list=PLqnslRFeH2UrcDBWF5mfPGpqQDSta6VK4&si=hWXNK549af8p-vHr
6. https://drive.google.com/file/d/1cjEMMoyDgi7qv5lJLm5b2kqlxr1EsL2B/view?usp=sharing
7. Chatgpt
8. W3schools