Tanya Chopra
102116069
4CS11
Email: tchopra1_be21@thapar.edu
SESS_LE1

# Report

**Paper summary** :The paper introduces the Speech Commands dataset, designed to advance keyword spotting systems by providing a standardized dataset of spoken words. This dataset allows researchers to build and test small models for detecting specific words amidst background noise. Released under the Creative Commons BY 4.0 license, it aims to support reproducible research and comparisons in keyword spotting, addressing the need for accessible and specialized datasets beyond traditional speech recognition resources.

**Statistical Analysis**:The dataset is divided into three sets, each containing 1000 samples:

- **Training Set:** Shows significant class imbalance, with `_unknown_` labels dominating at 63.2% of the samples. This imbalance could lead to the model being biased towards `_unknown_`, potentially compromising its performance on less frequent classes.
- **Validation Set:** Mirrors the training set's imbalance, with `_unknown_` labels making up 62.3% of the samples. This similarity suggests that the validation results may also reflect the same bias towards `_unknown_`.
- **Test Set:** Exhibits a more balanced distribution of classes, with `_unknown_` representing only 8.4% of the samples. This balance allows for a more accurate evaluation of the model's performance across different classes.

To address the imbalance in the training and validation sets, techniques such as class weighting, oversampling of minority classes, or undersampling of the majority class may be necessary.

## Model Performance Summary:

**Model Architecture:**

- **LSTM Layer (lstm_6):**
  - Output Shape: (None, 16000, 64)
  - Parameters: 16,896
- **LSTM Layer (lstm_7):**
  - Output Shape: (None, 64)
  - Parameters: 33,024

- **Dense Layer (dense_6):**
  - Output Shape: (None, 128)
  - Parameters: 8,320
- **Dropout Layer (dropout_3):**
  - Output Shape: (None, 128)
  - Parameters: 0
- **Dense Layer (dense_7):**
  - Output Shape: (None, 12)
  - Parameters: 1,548

The model "sequential_3" has a total of 59,788 parameters. During training:

- **Epoch 1:** Achieved 63.21% accuracy with a training loss of 1.5890; validation accuracy was 63.16% with a validation loss of 1.5280.
- **Epoch 2:** Accuracy slightly improved to 63.46% with a training loss of 1.5352; validation accuracy remained at 63.16% with a validation loss of 1.5312.
- **Epoch 3:** Accuracy slightly decreased to 63.39% with a training loss of 1.5327; validation accuracy stayed at 63.16% with a validation loss of 1.5270.

The model shows consistent performance with minor improvements in accuracy and loss.

he model "sequential_3" demonstrates stable performance with accuracy around 63% throughout training. Training and validation losses are closely aligned, indicating no significant overfitting. The model's ability to generalize is consistent across epochs, with minor variations in performance metrics. Further tuning or additional epochs might be needed to enhance model accuracy and reduce loss. Overall, the model performs reliably but shows room for improvement.

**Fine-tuning the model:**
To create a dataset for speech command recognition, I first organized a collection of audio samples, ensuring that each speech command was represented with 20 `.wav` files. This involved recording various commands such as "go," "next," and others, and then categorizing these recordings into labeled folders corresponding to each command. I processed this dataset by extracting the audio files from a ZIP archive, which contained subfolders for different commands. Each audio file was then loaded and preprocessed to match the input requirements of my model. This preprocessing included normalization and reshaping the audio data to a consistent format. To fine-tune the model, I planned to use this curated dataset by training the model specifically on these samples, adjusting the model parameters to improve its performance on recognizing speech commands from my dataset. This fine-tuning process aimed to enhance the model's accuracy and adaptability to my specific recordings, ensuring it could effectively classify each command based on the new data. (Training couldn’t happen due to GPU usage limits)

# New Dataset link:

https://drive.google.com/drive/folders/1YXucptgbDnMvhz3n545fxjtDldg0L-m4?usp=share_link

***And .ipynb file is part of the repository*