Speech Command Recognition Assignment Report

# Task 1: Dataset and Preprocessing

Dataset:  
For this task, I used Pytorch Commands dataset, which contains 1-second audio clips of spoken words. I focused on 10 specific command classes. The data was split as follows:  
Training set: 84,843 samples  
Validation set: 9,981 samples

## Feature Extraction:

I used MFCC (Mel-frequency cepstral coefficients) as the main audio feature, with the following configuration:  
Number of MFCCs: 40  
FFT window size: 1024  
Hop length: 500  
Number of Mel filters: 64

## Normalization and Padding:

To standardize the inputs:  
I computed the mean and standard deviation from a subset of the training data for normalization.  
All feature sequences were padded or truncated to a fixed length of 32 frames (MAX\_SEQ\_LEN), using zeros for padding.  
Normalization was done using the training statistics to ensure consistency.

## Challenges Faced:

The clips were of varying lengths, so padding and truncation were necessary to create uniform input sizes.  
Tuning the MFCC parameters to get meaningful features was a bit tricky and required several trials.  
There was a class imbalance issue in the dataset, which needed to be considered during training.

# Task 2: Simple RNN Classifier

## Model Overview:

Implemented a simple 2-layer RNN to classify MFCC features of shape (32 × 40).

## Model Architecture:

Input: MFCC features (32 frames × 40 coefficients)  
RNN Layer:  
Layers: 2  
Hidden size: 512  
Batch-first: True  
Activation: tanh (default)  
Output Layer:  
Fully connected Linear: 512 → 10 (classes)  
Regularization: None (no dropout or batchnorm)

## Training Setup:

Loss Function: CrossEntropyLoss  
Optimizer: Adam (lr = 1e-4)  
Epochs: 10  
Early Stopping: Patience = 3  
Batch Size: 64  
Device: GPU (if available), else CPU

## Training Performance:

Model showed consistent improvement over 10 epochs.  
Good convergence in later epochs, though performance saturates earlier than LSTM.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epoch | Train Loss | Val Loss | Train Acc | Val Acc |
| 1 | 1.7695 | 1.5901 | 0.3617 | 0.4564 |
| 2 | 1.4216 | 1.2337 | 0.5065 | 0.586 |
| 3 | 1.1367 | 1.0447 | 0.6084 | 0.6333 |
| 4 | 0.91 | 0.9018 | 0.6878 | 0.6962 |
| 5 | 0.7518 | 0.6849 | 0.7464 | 0.768 |
| 6 | 0.6511 | 0.6234 | 0.7847 | 0.7934 |
| 7 | 0.577 | 0.5633 | 0.8142 | 0.8234 |
| 8 | 0.4916 | 0.5257 | 0.8452 | 0.8374 |
| 9 | 0.4583 | 0.4761 | 0.8555 | 0.8512 |
| 10 | 0.4041 | 0.4286 | 0.8725 | 0.8682 |

## Test Performance:

Test Loss: 0.4716  
Test Accuracy: 86.01%

## Classification Report (RNN on Test Data):

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Precision | Recall | F1-score |
| yes | 0.88 | 0.95 | 0.92 |
| no | 0.78 | 0.66 | 0.71 |
| up | 0.88 | 0.81 | 0.84 |
| down | 0.77 | 0.88 | 0.82 |
| left | 0.92 | 0.83 | 0.87 |
| right | 0.96 | 0.89 | 0.92 |
| on | 0.83 | 0.92 | 0.88 |
| off | 0.91 | 0.88 | 0.89 |
| stop | 0.93 | 0.96 | 0.94 |
| go | 0.75 | 0.83 | 0.79 |

# Task 3: LSTM Classifier

## Model Overview:

Built an LSTM-based model to classify MFCC sequences (shape: 32 × 40).

## Model Architecture:

Input dimension: 40 (MFCC features)  
Hidden dimension: 128  
Layers: 2  
Bidirectional: Yes  
Dropout: 0.5  
Output dimension: 10 (command classes)  
Final layer: Linear(hidden\_dim × 2 → 10)

## Training Setup:

Loss function: CrossEntropyLoss  
Optimizer: Adam (lr = 1e-3)  
Epochs: 10  
Batch size: 64  
Device: GPU (if available), else CPU

## Training Performance:

Train and validation accuracy consistently improved over epochs.  
Early signs of convergence by ~epoch 8–9.  
LSTM outperformed simple RNN on both accuracy and stability.

## Test Performance:

Test Loss: 0.3226  
Test Accuracy: 91.95%

## Classification Report (LSTM on Test Data):

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Precision | Recall | F1-score |
| yes | 0.96 | 0.98 | 0.97 |
| no | 0.91 | 0.85 | 0.88 |
| up | 0.89 | 0.93 | 0.91 |
| down | 0.92 | 0.87 | 0.9 |
| left | 0.9 | 0.94 | 0.92 |
| right | 0.97 | 0.93 | 0.95 |
| on | 0.93 | 0.93 | 0.93 |
| off | 0.93 | 0.87 | 0.9 |
| stop | 0.97 | 0.98 | 0.97 |
| go | 0.83 | 0.91 | 0.87 |

# LSTM vs RNN – Performance Comparison for Speech Command Recognition

The LSTM model significantly outperforms the RNN in every key evaluation metric. LSTM achieved a test accuracy of 91.95%, with a test loss of 0.3226, a macro F1-score of 0.92, and a weighted F1-score of 0.92. In contrast, the RNN achieved a test accuracy of 86.01%, with a test loss of 0.4716, and both macro and weighted F1-scores of 0.86.  
  
This clearly shows that LSTM handles sequential audio features more effectively than a simple RNN, leading to better generalization and precision across commands.

# Conclusion

LSTM provides superior performance over RNN for speech command recognition tasks.