Experiment Report: Speech Commands Classification with Conv1D and Grouped Convolutions

Your Name

March 29, 2025

1 Introduction

This report presents an experimental study on classifying speech commands using a 1D convolutional neural network (CNN). The focus of the experiments was to evaluate the impact of varying the number of mel filterbanks (`n_mels`) and the groups parameter in grouped convolutions on model performance. Metrics such as model parameters, floating-point operations (FLOPs), and validation accuracy were analyzed to determine the optimal configuration.

The dataset used was derived from the `SPEECHCOMMANDS` dataset, preprocessed into log-mel spectrograms. Experiments were conducted for different combinations of `n_mels` (20, 40, 80) and `groups` (1, 2, 4, 8, 16).

2 Methodology

2.1 Model Architecture

The model architecture consisted of:
- An initial 1D convolutional layer to map input channels to 32.
- A grouped 1D convolutional layer with adjustable `groups`.
- Max-pooling layers for downsampling.
- Fully connected layers for classification.

2.2 Input Preprocessing

Log-mel spectrograms were generated using the `LogMelFilterBanks` module with varying numbers of mel filterbanks (`n_mels`). Each spectrogram had dimensions (`n_mels`, time_frames).

2.3 Evaluation Metrics

The following metrics were logged for each experiment:
- **Parameters**: Number of trainable parameters in the model.
- **FLOPs**: Computational cost measured in floating-point operations.
- **Validation Accuracy**: Final accuracy on the validation set.

3 Results

The results of the experiments are summarized in Table 1. Each row corresponds to a specific combination of `n_mels` and `groups`.

Table 1: Summary of Results

n_mels	Groups	Parameters	FLOPs	Final Accuracy
20	1	204,866	568,448.0	0.9649
20	2	203,330	416,384.0	0.9731
20	4	202,562	340,352.0	0.9693
20	8	202,178	302,336.0	0.9612
20	16	201,986	283,328.0	0.9768
40	1	205,506	631,808.0	0.9724
40	2	203,970	479,744.0	0.9706
40	4	203,202	403,712.0	0.9756
40	8	202,818	365,696.0	0.9612
40	16	202,626	346,688.0	0.9750
80	1	206,786	758,528.0	0.9136
80	2	205,250	606,464.0	0.9549
80	4	204,482	530,432.0	0.9643
80	8	204,098	492,416.0	0.9750
80	16	203,906	473,408.0	0.9493

4 Graphical Analysis

To visualize the results, several plots were generated:

4.1 Train Loss vs Epoch

The first graph shows the training loss over epochs for different configurations of n_mels and groups. Lower training loss indicates better convergence during training.

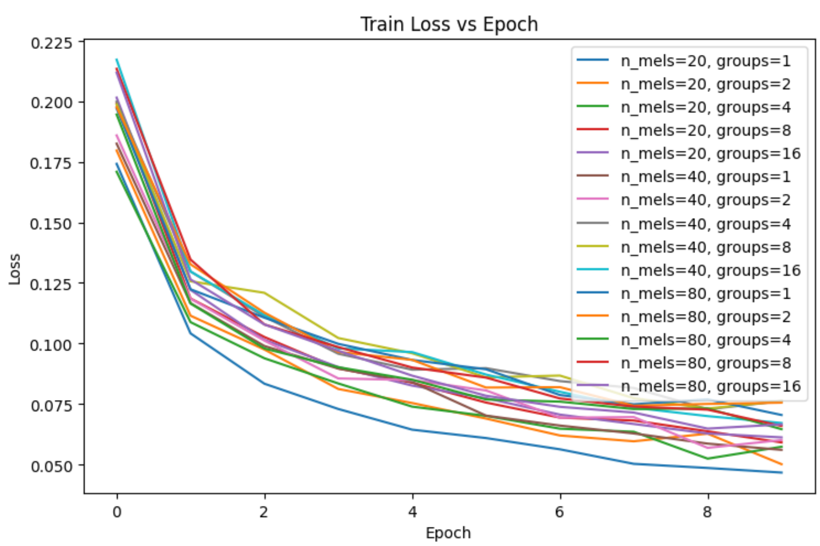

Figure 1: Training Loss vs Epoch for Different Configurations

4.2 Validation Accuracy vs Epoch

The second graph illustrates the validation accuracy over epochs. Higher accuracy indicates better generalization to unseen data.

4.3 Epoch Time vs Groups

The third graph compares the epoch training time for different values of groups, grouped by n_mels. This highlights the trade-off between computational efficiency and model complexity.

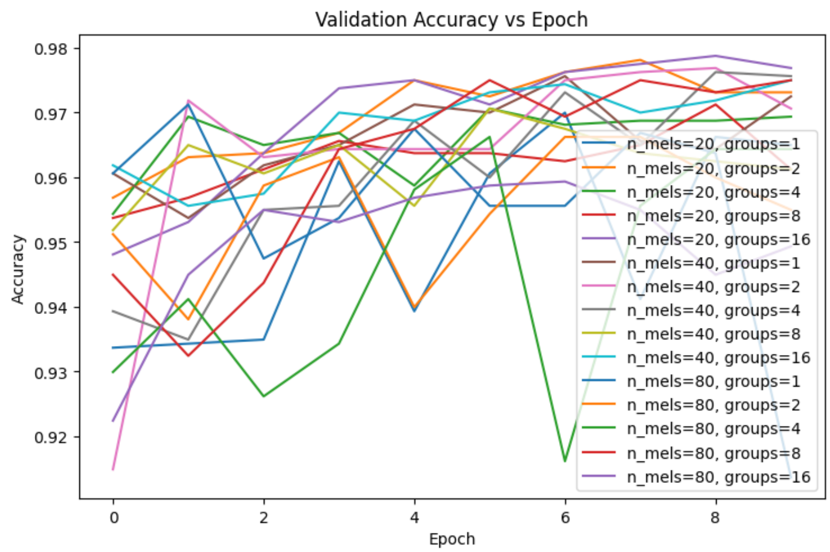

Figure 2: Validation Accuracy vs Epoch for Different Configurations

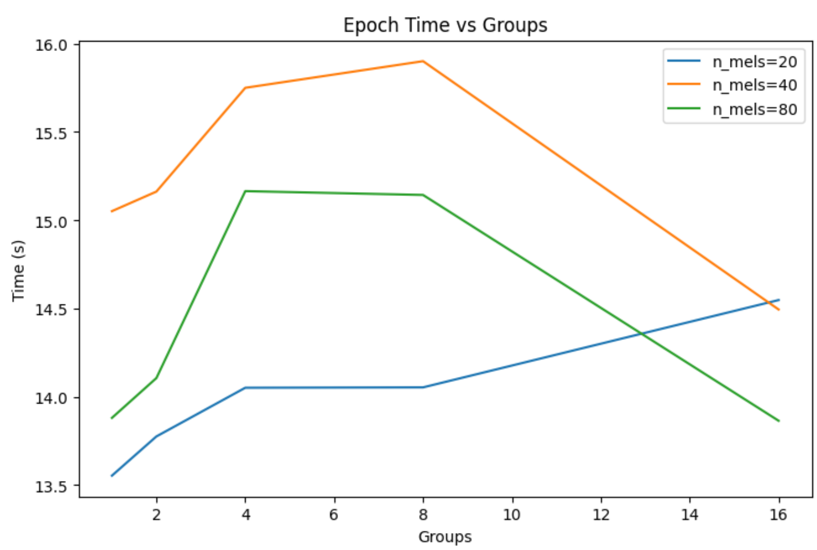

Figure 3: Epoch Training Time vs Groups for Different n_mels

4.4 Testing Accuracy vs n_mels

The fourth graph shows the relationship between testing accuracy and `n_mels`. It provides insights into how increasing the number of mel filterbanks affects model performance.

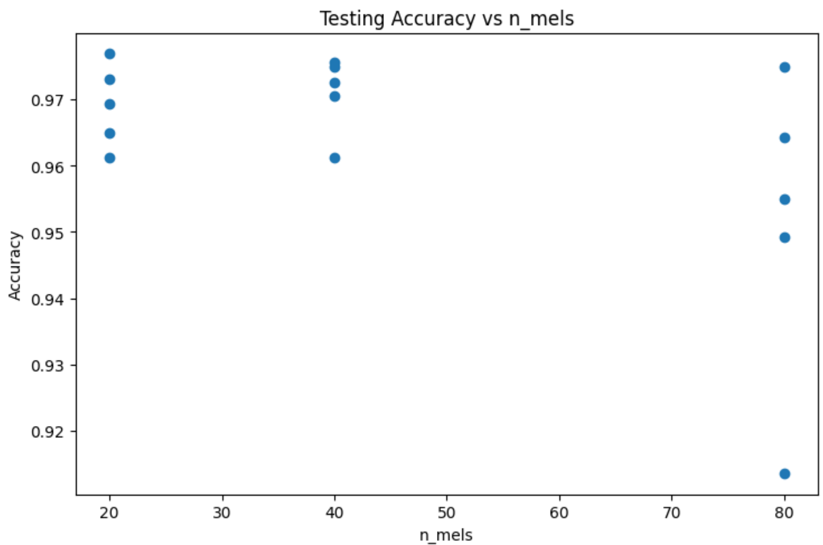

Figure 4: Testing Accuracy vs n_mels

5 Analysis

5.1 Impact of Groups

Increasing the `groups` parameter reduced both the number of parameters and FLOPs, as expected. However, the effect on validation accuracy varied:

- For `n_mels=20`, higher `groups` values (e.g., 16) achieved the best accuracy (0.9768).

- For `n_mels=40`, moderate `groups` values (e.g., 4) performed well (0.9756).

- For `n_mels=80`, higher `groups` values (e.g., 8) achieved the best accuracy (0.9750).

5.2 Impact of n_mels

Increasing `n_mels` increased both parameters and FLOPs but did not consistently improve accuracy:

- `n_mels=20` generally performed better than `n_mels=80`.

- `n_mels=40` offered a good balance between computational cost and accuracy.

5.3 Optimal Configuration

The best-performing configuration was:

- `n_mels=20`, `groups=16`: Final Accuracy = 0.9768, Parameters = 201,986, FLOPs = 283,328.0.

6 Conclusions

This study demonstrated the trade-offs between model complexity (parameters and FLOPs) and performance (validation accuracy) when varying `n_mels` and `groups`. Key findings include:

- Higher `groups` values reduce computational cost while maintaining or improving accuracy.

- Moderate `n_mels` values (e.g., 20 or 40) achieve better performance than very high values (e.g., 80).

- The optimal configuration (`n_mels=20`, `groups=16`) achieved the highest accuracy with relatively low computational cost.

Future work could explore additional architectural modifications, such as deeper networks or attention mechanisms, to further improve performance.