

## Machine-Learning Project (Fall 2019)

### PRE-PROCESSING

MFCC: Zero padding, Normalizing, Standardizing

Labels: Vectorized and one-hot encoded

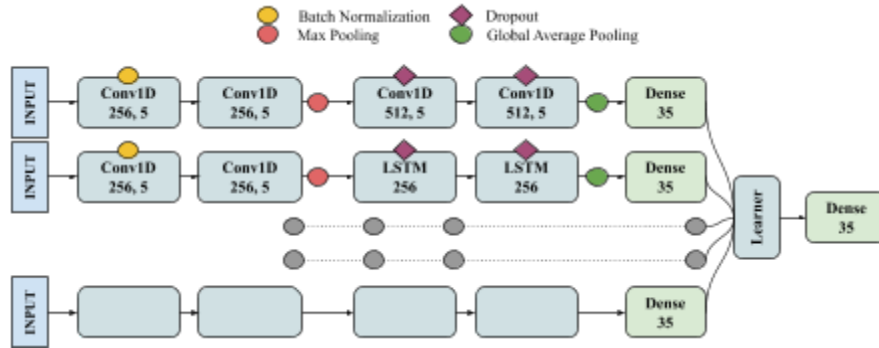
Upsampling has been applied to address the imbalanced data problem. Normalizing and standardizing did not show significant improvement, so we did not apply them.

### DATA AUGMENTATION

Based on the concept of better generalizability through more data, we applied several data augmentation methods on the MFCC data such as time and frequency masking to create synthetic data to add more variation to the training data.

### LEARNING ALGORITHM

We began our experimentations by training an MLP and fine-tuned the model and then shifted to use a 1d-Convolutional NN as our base architecture. Conv1D with LSTM layers on top was introduced later as we expected the LSTM layers to learn spatial relationships as sequences, while we assume the first convolutional layer functions as a feature extractor for the following LSTM layer. We selected models for ensemble and stacking based on their performance on multiple confusion matrices. We could see that different models have difficulty predicting classes that are similar, for instance, “three” and “tree” and “forward” and “follow”, so we added a bidirectional wrapper on the LSTM layer, assuming learning sequence features in a reversed order might help the LSTM to learn patterns at the end of sequences better.



	Model	Augmentation	Dropout	Optimizer	Lr	Batch-size	Epoch	Validation	Test	Comments
1	Conv1D	No augmentation	0.4	Adam	0.01	256	48	94.66	93.69	-
2	Conv1D	Upsampling	0.35 / 0.5	Adam	2e-3	128	40	95.14	94.93	Data Perm
3	Conv1D	Upsam + Time masking	0.35 / 0.5	Adam	2e-3	128	20	95.46	95.26	Data Perm
4	Stacked-7	Upsam + Time&Freq mask	0.10	Adam	0.001	mixed	25	95.65	95.85	Data Perm
5	Stacked-4	Upsam + random augment	0.10	Adam	0.001	mixed	20	96.10	95.45	Data Perm
6	Stacked-16	Upsam + random augment	mixed	Adam	mixed	mixed	mixed	96.45	96.31	Data Perm
7	Stacked-18	Upsam + random augment	mixed	Adam	mixed	mixed	mixed	96.47	96.31	Data Perm