Bosch: voice deep learning engineer, test-task report

Candidate name: Daria Diatlova,
GitHub repository: https://github.com/dariadiatlova/sound-classification

Problem statement

10-class classification, UrbanSound dataset.

Model: LSTM:

- 4 layers;
- hidden dim: 256
- 1.9M parameters;
- Dropout: 0.25.

Audio data features:

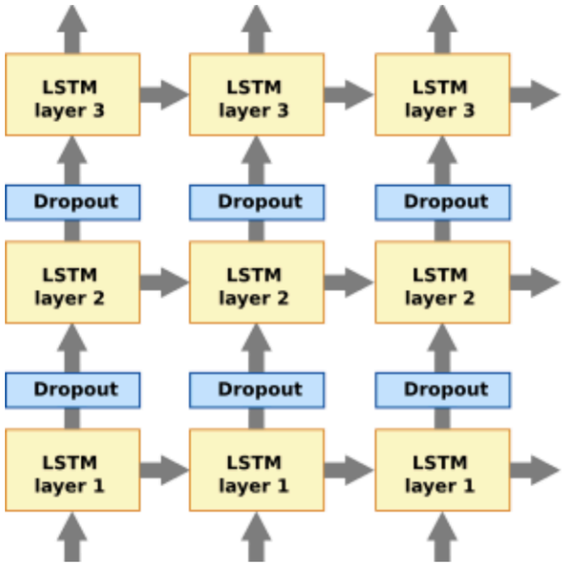
- sample rate 44100;
- 1 channel;
- 4 seconds length;
- mel spectrogram (1024 FFT, 64 mels), db;
- augmentation:
  - frequency and time masking: 10%, spec.mean().

Train and Validation setup

Weights & Biases

PyTorch Lightning

PyTorch



10 fold cross-validation (default division – UrbanSound).

Training length: 50 epochs.

Model Architecture like on the picture + 4th LSTM layers and FC on 10 classes in the end.

Each cell takes hidden and cell state, for the first cell h0 and c0 are empty.

Loss Criterion: CrossEntropy.

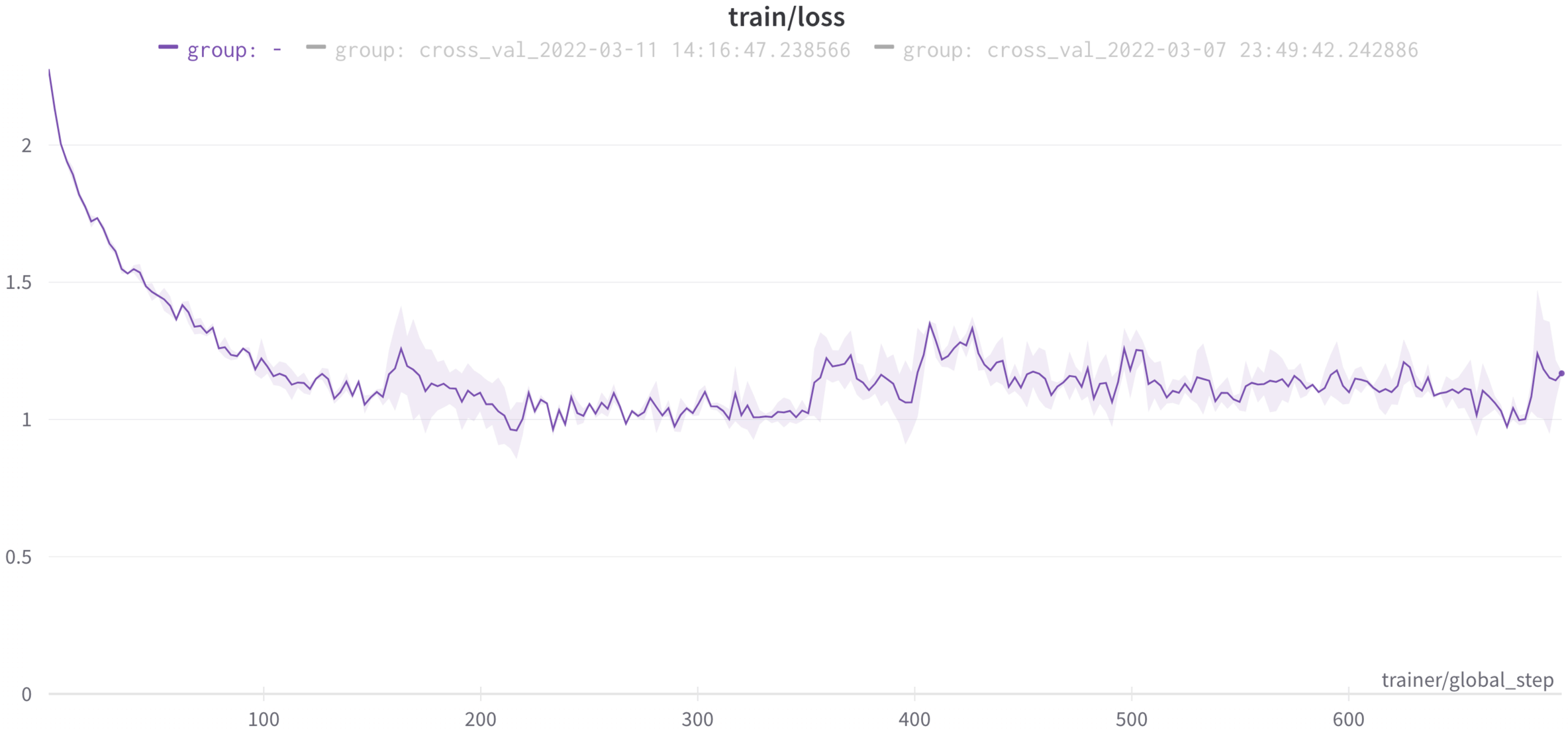
Dropout=0.25, disabled during evaluation.

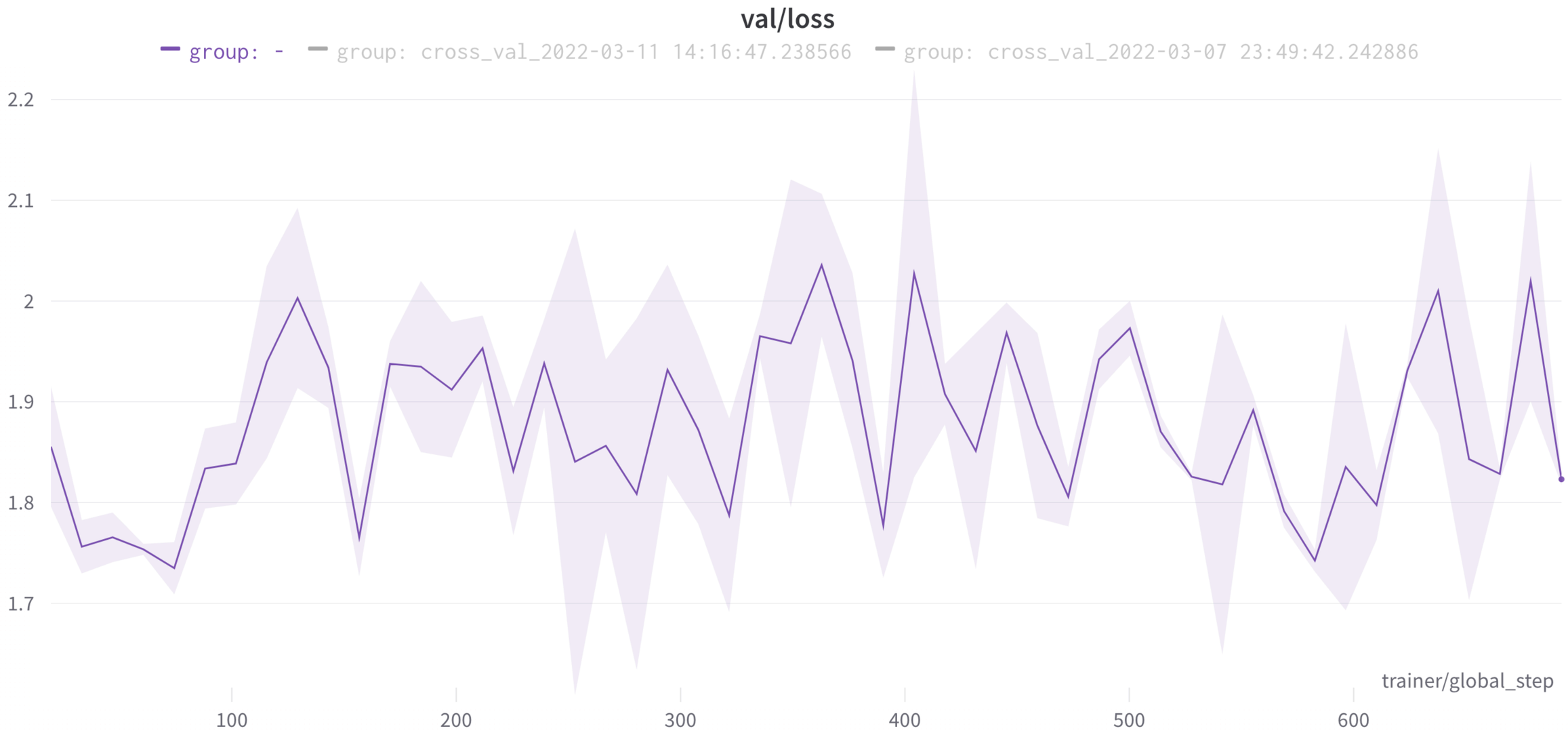
Plots*

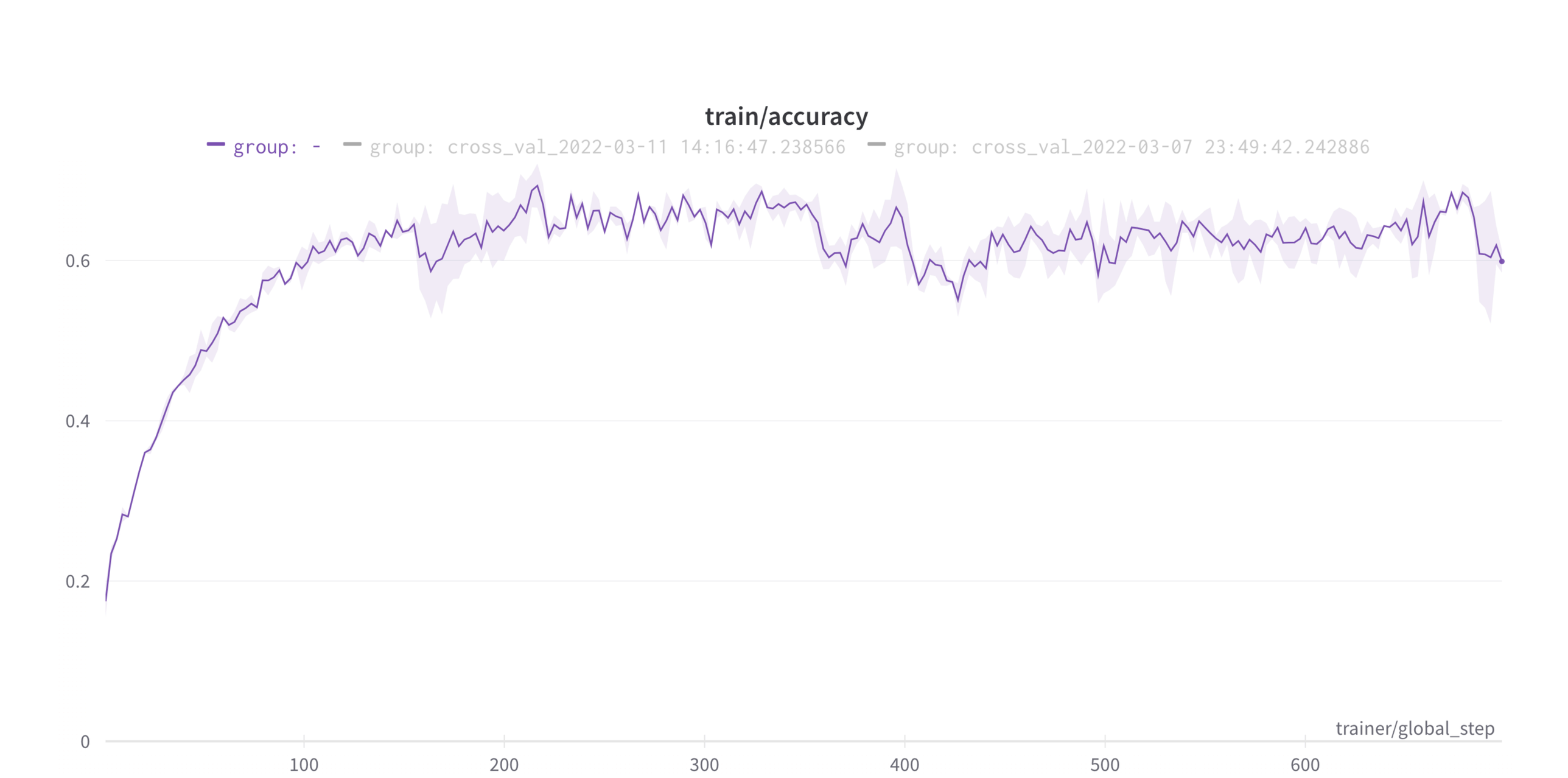
***** Due to the limitation of computing resources, the result plots is an average of 2 fold cross validation:

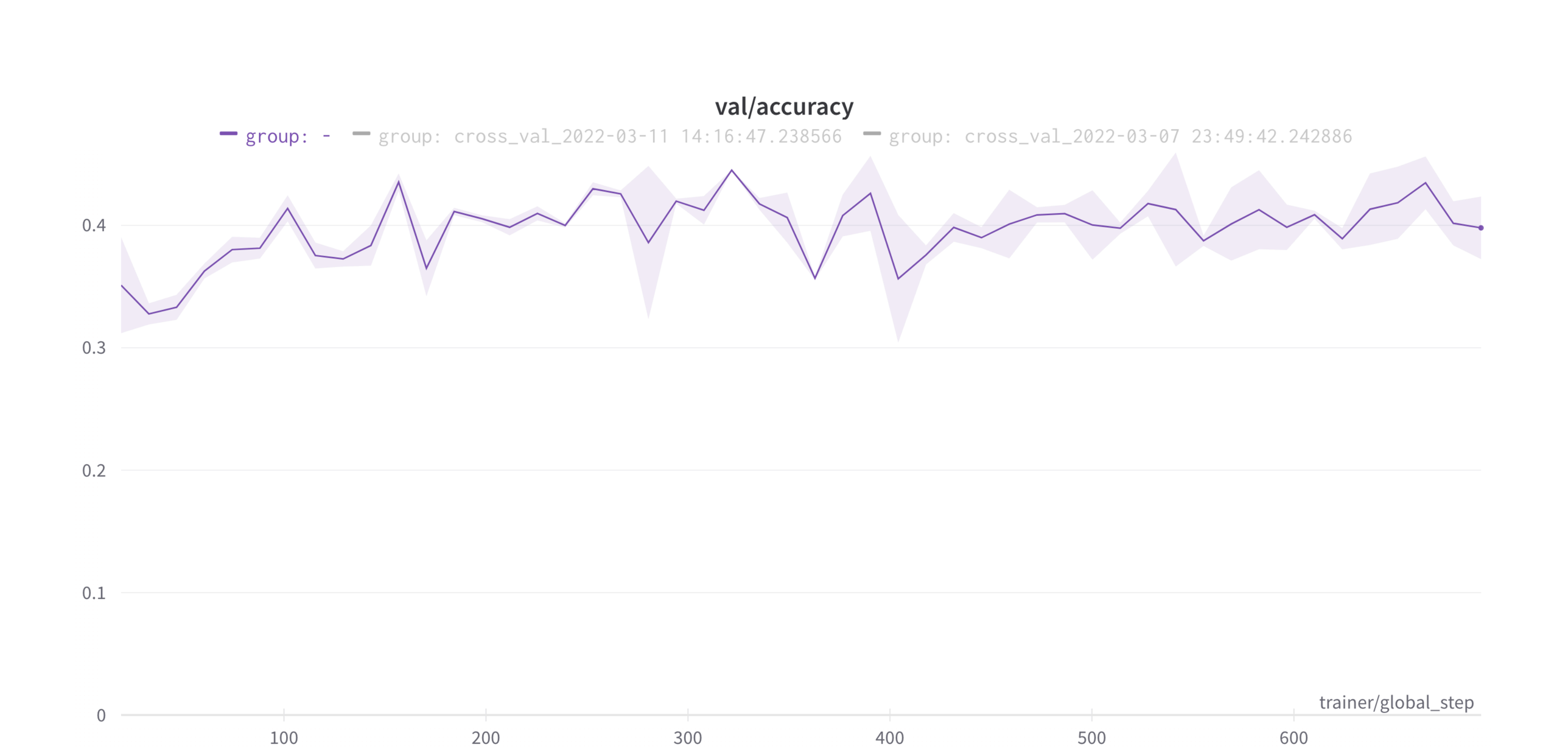
- train 50 epochs on 2-10 folders and evaluate on 1st folder;
- train 50 epochs on 1, 3-10 folders and evaluate on 2nd folder.

P.S.: takes to long to repeat the process 10 times without gpu :(


train/loss
group: -
group: cross_val_2022-03-11 14:16:47.238566
group: cross_val_2022-03-07 23:49:42.242886
2
1.5
1
0.5
0
100
200
300
400
500
600
trainer/global_step


val/loss
group: -
group: cross_val_2022-03-11 14:16:47.238566
group: cross_val_2022-03-07 23:49:42.242886
2.2
2.1
2
1.9
1.8
1.7
trainer/global_step
100
200
300
400
500
600


train/accuracy
group: -
group: cross_val_2022-03-11 14:16:47.238566
group: cross_val_2022-03-07 23:49:42.242886
0.6
0.4
0.2
0
100
200
300
400
500
600
trainer/global_step


val/accuracy
group: -
group: cross_val_2022-03-11 14:16:47.238566
group: cross_val_2022-03-07 23:49:42.242886
0.4
0.3
0.2
0.1
0
100
200
300
400
500
600
trainer/global_step

Analysis of results

In the provided solution LSTM model with 1.9M parameters, 4 layers and dropout 0.25 was used.

The best result achieved: accuracy score 0.68, 0.44 on train / val datasets.

Things I consider important to do but didn't do due to the time-constraint.

1. Run CNN model provided in tutorial to have a baseline to compare with;
2. Experiment with:
   - augmentation params, input data characteristics (different window size, mfcc, stack of features);
   - lstm params: number of layers, hidden dim size, dropout (smaller, bigger deactivate between layers, activate before last layer).