PADERBORN UNIVERSITY

NT

Deep Learning for Speech and Audio

Running Large-Scale Experiments

Christoph Boeddeker, Tobias Cord-Landwehr, Reinhold Haeb-Umbach

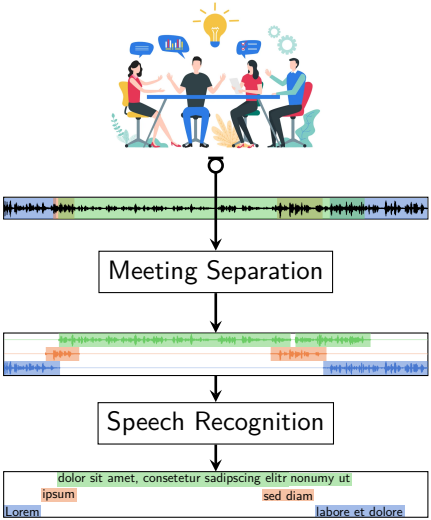
Paderborn University, Department of Communications Engineering, Germany

Motivation: Meeting transcription (Part 1)

Source: https://www.strety.com/blog/run-an-effective-team-meeting

- Process a meeting recording
- Create transcription
- Applications:
  - Automatic protocolling
  - Smart home assistants

Motivation: Meeting transcription (Part 2)



- Meeting transcription
  - Meeting separation ← my research
    - Who spoke when?
    - Separate overlapping speech
    - Remove noise
  - Speech Recognition
    - Audio to text
- Techniques
  - Neural networks (NN)
  - Statistical Models
  - Combinations
- Realization aspects ← today
  - How do we use Noctua2?

Neural Network (NN) training

- Training
  - Load a few examples
  - Prepare the data
  - Apply NN and compute loss
  - Update NN parameters
  - Repeat with other examples until convergence

- Difficulties
  - Database does not fit into memory (e.g. 250 GB)
    - Prepare data as it is required (on-demand)
    - Preprocessing takes time and increases the size (e.g. 3000 GB)
    - Prefetching: Use multiple threads to load examples and prepare examples before they are requested
  - Parallelization issue
    - NN parameter update required before next "apply"
    - Utilize GPUs for speedup (Most frameworks have CUDA support)

Neural Network (NN) training

- Training
  - Load a few examples
  - Prepare the data
  - Apply NN and compute loss
  - Update NN parameters
  - Repeat with other examples until convergence

- Difficulties
  - Database does not fit into memory (e.g. 250 GB)
    - Prepare data as it is required (on-demand)
    - Preprocessing takes time and increases the size (e.g. 3000 GB)
    - Prefetching: Use multiple threads to load examples and prepare examples before they are requested
  - Parallelization issue
    - NN parameter update required before next "apply"
    - Utilize GPUs for speedup (Most frameworks have CUDA support)

Neural Network (NN) training

- Training
  - Load a few examples
  - Prepare the data
  - Apply NN and compute loss
  - Update NN parameters
  - Repeat with other examples until convergence

- Difficulties
  - Database does not fit into memory (e.g. 250 GB)
    - Prepare data as it is required (on-demand)
    - Preprocessing takes time and increases the size (e.g. 3000 GB)
    - Prefetching: Use multiple threads to load examples and prepare examples before they are requested
  - Parallelization issue
    - NN parameter update required before next "apply"
    - Utilize GPUs for speedup (Most frameworks have CUDA support)

Neural Network (NN) training

- Training
  - Load a few examples
  - Prepare the data
  - Apply NN and compute loss
  - Update NN parameters
  - Repeat with other examples until convergence

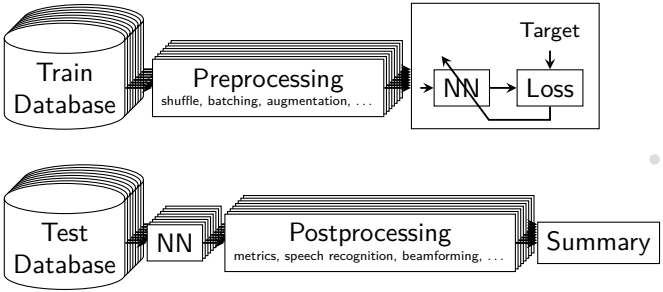
- Difficulties
  - Database does not fit into memory (e.g. 250 GB)
    - Prepare data as it is required (on-demand)
    - Preprocessing takes time and increases the size (e.g. 3000 GB)
    - Prefetching: Use multiple threads to load examples and prepare examples before they are requested
  - Parallelization issue
    - NN parameter update required before next “apply”
    - Utilize GPUs for speedup (Most frameworks have CUDA support)

Neural Network (NN): Inference

- Inference (apply to thousands of examples)
  - Apply NN
  - Postprocessing
    - Expensive and several “CPU only” limitations
    - e.g. Metric calculation, speech recognition, statistical models, beamforming, ...
  - Summarization, e.g. mean metric

- Embarrassingly parallel workload:
  - Message Passing Interface (MPI)
    - Alternative to Thread/Process pools and SLURM array jobs
    - Scales over node boundaries
  - Close to no communication between workers
  - dlp_mpi: Our thin wrapper of mpi4py
  - mpi4py: Python bindings for MPI

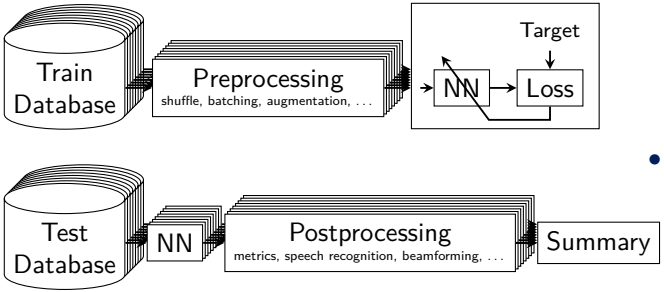
Neural Network (NN): Inference



- Inference (apply to thousands of examples)
  - Apply NN
  - Postprocessing
    - Expensive and several “CPU only” limitations
    - e.g. Metric calculation, speech recognition, statistical models, beamforming, ...
  - Summarization, e.g. mean metric

- Embarrassingly parallel workload:
  - Message Passing Interface (MPI)
    - Alternative to Thread/Process pools and SLURM array jobs
    - Scales over node boundaries
  - Close to no communication between workers
  - dlp_mpi: Our thin wrapper of mpi4py
  - mpi4py: Python bindings for MPI

Neural Network (NN): Inference

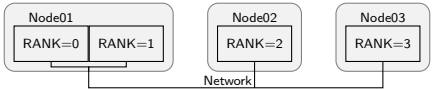


- Inference (apply to thousands of examples)
  - Apply NN
  - Postprocessing
    - Expensive and several "CPU only" limitations
    - e.g. Metric calculation, speech recognition, statistical models, beamforming, ...
  - Summarization, e.g. mean metric

- Embarrassingly parallel workload:
  - Message Passing Interface (MPI)
    - Alternative to Thread/Process pools and SLURM array jobs
    - Scales over node boundaries
  - Close to no communication between workers
  - dlp_mpi: Our thin wrapper of mpi4py
  - mpi4py: Python bindings for MPI

Embarrassing parallelism with MPI: Introduction

- Message Passing Interface (MPI)
  - My view:
    - Launch “SIZE” processes (scattered on N nodes)
    - Unique “RANK” for each process
    - Provides process communication tools (send, receive, broadcast, gather, ...)
  - Scales from notebook over workstations to HPC systems



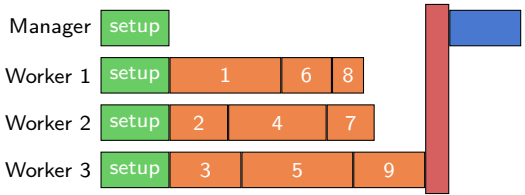
Embarrassing parallelism with MPI: Problem

```
filelist = ...
results = []
for file in filelist:
    result = metric(NN(load(file)))
    results.append(result)
dump(results)
```

python inference.py

Embarrassing parallelism with MPI: Scheduling

```
filelist = ...
results = dlp_mpi.collection.UnorderedList()
for file in dlp_mpi.split_managed(filelist):
    result = metric(NN(load(file)))
    results.append(result)
results = results.gather()
if dlp_mpi.IS_MASTER:
    dump(results)
```



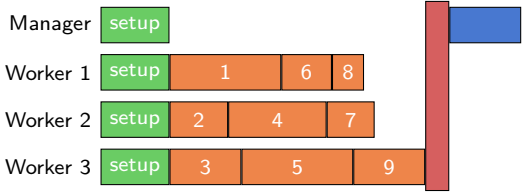
srun ... -n 2 -t 30:00 python inference.py

Open Source: https://pypi.org/project/dlp-mpi

Embarrassing parallelism with MPI: Scheduling

```
filelist = ...
results = dlp_mpi.collection.UnorderedList()
for file in dlp_mpi.split_managed(filelist):
    result = metric(NN(load(file)))
    results.append(result)
results = results.gather()
if dlp_mpi.IS_MASTER:
    dump(results)
```

MPI related code



srun ... -n 2 -t 30:00 python inference.py

Open Source: `https://pypi.org/project/dlp-mpi`

Summary

- Different requirements for training and inference
- Training
  - Single node
  - GPU-intensive
- Inference (Includes model-based approaches):
  - Easily scalable (multi-node)
  - CPU-intensive
- 20 department publications that profited from PC^2 infrastructure over the last two years

Papers with PC^2 Acknowledgement

[1] Segment-Less Continuous Speech Separation of Meetings: Training and Evaluation Criteria T. von Neumann, K. Kinoshita, C. Boeddeker, M. Delcroix, R. Haeb-Umbach, IEEE/ACM TASLP 31 (2023) 576–589.
[2] Frame-Wise and Overlap-Robust Speaker Embeddings for Meeting Diarization T. Cord-Landwehr, C. Boeddeker, C. Zoril, R. Doddipatla, R. Häb-Umbach, in: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2023.
[3] A Teacher-Student Approach for Extracting Informative Speaker Embeddings From Speech Mixtures T. Cord-Landwehr, C. Boeddeker, C. Zoril, R. Doddipatla, R. Häb-Umbach, in: INTERSPEECH 2023, ISCA, 2023.
[4] MeetEval: A Toolkit for Computation of Word Error Rates for Meeting Transcription Systems T.C. von Neumann, C. Boeddeker, M. Delcroix, R. Haeb-Umbach, in: Proc. CHiME 2023 Workshop on Speech Processing in Everyday Environments, 2023.
[5] Neural Network Based Carrier Frequency Offset Estimation From Speech Transmitted Over High Frequency Channels J. Heitkämper, J. Schmalenstroeer, R. Haeb-Umbach, in: Proceedings of the 30th European Signal Processing Conference (EUSIPCO), Belgrad, n.d.
[6] An Initialization Scheme for Meeting Separation with Spatial Mixture Models C. Boeddeker, T. Cord-Landwehr, T. von Neumann, R. Haeb-Umbach, in: Interspeech 2022, ISCA, 2022.
[7] MMS-MSG: A Multi-purpose Multi-Speaker Mixture Signal Generator T. Cord-Landwehr, T. von Neumann, C. Boeddeker, R. Haeb-Umbach, in: IWAENC, 2022.
[8] SA-SDR: A Novel Loss Function for Separation of Meeting Style Data T. von Neumann, K. Kinoshita, C. Boeddeker, M. Delcroix, R. Haeb-Umbach, in: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022.
[9] Monaural source separation: From anechoic to reverberant environments T. Cord-Landwehr, C. Boeddeker, T. von Neumann, C. Zorila, R. Doddipatla, R. Haeb-Umbach, in: 2022 International Workshop on Acoustic Signal Enhancement (IWAENC), IEEE, Bamberg, 2022.
[10] Informed vs. Blind Beamforming in Ad-Hoc Acoustic Sensor Networks for Meeting Transcription T. Gburrek, J. Schmalenstroeer, J. Heitkaemper, R. Haeb-Umbach, in: IWAENC, IEEE, 2022.
[11] A Meeting Transcription System for an Ad-Hoc Acoustic Sensor Network T. Gburrek, C. Boeddeker, T. von Neumann, T. Cord-Landwehr, J. Schmalenstroeer, R. Haeb-Umbach, A Meeting Transcription System for an Ad-Hoc Acoustic Sensor Network, arXiv, 2022.
[12] Investigation into Target Speaking Rate Adaptation for Voice Conversion M. Kuhlmann, F. Seebauer, J. Ebbers, P. Wagner, R. Häb-Umbach, in: Interspeech 2022, ISCA, 2022.
[13] Far-Field Automatic Speech Recognition R. Haeb-Umbach, J. Heymann, L. Drude, S. Watanabe, M. Delcroix, T. Nakatani, Proceedings of the IEEE 109 (2021) 124–148.
[14] Convolutive Transfer Function Invariant SDR Training Criteria for Multi-Channel Reverberant Speech Separation C. Boeddeker, W. Zhang, T. Nakatani, K. Kinoshita, T. Ochiai, M. Delcroix, N. Kamo, Y. Qian, R. Haeb-Umbach, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021.
[15] Contrastive Predictive Coding Supported Factorized Variational Autoencoder for Unsupervised Learning of Disentangled Speech Representations J. Ebbers, M. Kuhlmann, T. Cord-Landwehr, R. Haeb-Umbach, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 3860–3864.
[16] Speeding Up Permutation Invariant Training for Source Separation T. von Neumann, C. Boeddeker, K. Kinoshita, M. Delcroix, R. Haeb-Umbach, in: Speech Communication; 14th ITG Conference, 2021.
[17] Self-Trained Audio Tagging and Sound Event Detection in Domestic Environments J. Ebbers, R. Haeb-Umbach, in: Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCA), Barcelona, Spain, 2021, pp. 226–230.
[18] Adapting Sound Recognition to A New Environment Via Self-Training J. Ebbers, M.C. Keyser, R. Haeb-Umbach, in: Proceedings of the 29th European Signal Processing Conference (EUSIPCO), 2021, pp. 1135–1139.
[19] A Database for Research on Detection and Enhancement of Speech Transmitted over HF links J. Heitkaemper, J. Schmalenstroeer, V. Ion, R. Haeb-Umbach, in: Speech Communication; 14th ITG-Symposium, 2021, pp. 1–5.
[20] Graph-PIT: Generalized Permutation Invariant Training for Continuous Separation of Arbitrary Numbers of Speakers T. von Neumann, K. Kinoshita, C. Boeddeker, M. Delcroix, R. Haeb-Umbach, in: Interspeech 2021, 2021.