Hands-On Assignment

Introduction to Deep Learning
800883-B-6

Spring, 2025

Objective

In this assignment, your goal is to develop and compare deep learning models for classifying spoken sentences in a given language into accents corresponding to a geographical area. The dataset consists of audio recordings of individuals uttering simple sentences with accents from different countries. Your task is to train neural networks that can effectively classify these accents based on the audio input.

Description

This is a group assignment. The groups should consist of 4 to 5 people. You must implement, train, and evaluate models. You must submit a report describing your solution(s) and the code you used. Your model's performance will be compared to that of a baseline model on a provided hold-out test set.

This assignment is worth 30% of your course grade. The assignment grade will be based on the quality of your work as judged by the instructors based on your **report** and **code** following a specific rubric.

In addition, you will get a bonus based on your ranking on a shared task competition leaderboard. Specifically:

- if your group ranks first, you will receive a bonus of 2 points on the assignment's grade;
- if your score is no better than a provided baseline, you will receive no bonus;
- for intermediate ranks between the best and the baseline scores, the bonus points will be linearly interpolated. The performance of the baseline solution will be shown on the competition's website.

Later, a separate document will specify all the information about the competition and evaluation on the test set.

Passing the assignment is not mandatory for passing the course. Still, it is highly advisable to submit your solutions, as not doing so implies a 0 in 30% of your final grade (there is no

compensation with the exam). There will be no resit for this assignment as passing it is not compulsory, and the course can be passed without passing the assignment. The exam may include questions that might be easier to answer if you have worked on the shared task.

1 Shared Task – Accent classification of Spoken Language using Deep Learning

The data set you will work with for this assignment is a collection of utterances recorded in a controlled environment. This data set is rich and diverse, featuring utterances from multiple male and female speakers from different countries. The sentences are in the same language but with five different accents. Each utterance is stored in a .wav file and is annotated with the corresponding accent and gender of the speaker.

1.1 Data format

Here are the key characteristics of the dataset:

- **Utterances:** The dataset comprises various utterances from different actors. They are recorded with a sampling rate of 16000 and are single channel.
- **Annotations:** Each audio file is annotated with the corresponding accent and the speaker's gender. The accent is encoded in the first character of the file name with a single number from 1 to 5. The gender is encoded as a single letter ('m' or 'f') corresponding to the second character of the file name.

Each recording varies in length, so your model must be capable of handling inputs of different lengths. The data set will be provided in a ZIP file containing a folder with all recordings in .wav format. As explained above, annotations are embedded in the file names.

Your final model must be evaluated on a given test set that will be provided later. The test set only contains the recordings, so no annotations are given. You have to submit your classification results to a competition website to get your model's performance on that set.

Your goal is to build a Deep Learningmodel that is capable of estimating the accent of a recording. Understanding your dataset is the first step towards creating a successful model. So, take your time to explore and understand the data.

1.2 Required experiments

To achieve the goals of this assignment, you must ensure you complete the following:

1. Model Comparisons:
    - Train and compare the performance of (at least) these two approaches:
        (a) Use the raw input signal and analyze it as a 1D input; in this case, you can, of course, apply simple pre-processing like standardization.
        (b) Use a transformation or feature engineering and use the transformed version as the input to the model. In this case, the format of the signal can be different from the original one (e.g., Short Time Fourier Transform).

- For each approach, you can use different deep learning architectures (e.g., Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Transformers, etc.).
- Analyze and discuss the strengths and weaknesses of each approach and architecture.

2. Regularization Techniques: Implement and evaluate the impact of regularization techniques such as dropout, batch normalization, or weight decay. Explain how these techniques influence model performance and generalization.

3. Performance Evaluation: Assess the classification performance of your models using appropriate metrics such as accuracy, precision, recall, and F1-score. Additionally, analyze performance regarding the following:

   - By class (accent): Determine if the model performs better on certain accents than others and explore potential reasons.
   - By speaker's gender: Investigate whether gender biases exist in model predictions.

4. Data Augmentation: Implement data augmentation techniques that match your approach. Evaluate how these techniques impact model performance and whether they improve generalization.

2 Deliverables

You must submit a report and the code reflecting experiments reported and used to arrive at the reported results. **A submission without any of the two elements (report and code) is considered incomplete and will not be graded.**

2.1 Report

A report should be submitted by May 23rd, 2025. This report should follow a specific format and include the following:

- **Format:** The report must be correctly formatted and only contain three pages as follows:

  1. A title page lists the group members' names, student codes, and the group number.
  2. Two content pages (appendices are allowed, but the grade is based on the content page). Please note that anything outside the content pages will not be graded.

- The content pages must include the following:

  1. **Data Preprocessing:** The report should contain a concise section detailing the data pre-processing steps. It should provide adequate reasoning and arguments for the design decisions made during this process.
  2. **Experiments:** The report should briefly describe the experiments conducted and reported. This must be in line with the results reported and include:
     - Straightforward description of the two approaches evaluated.
     - Regularization techniques applied.
     - Relevant training settings such as hyperparameter tuning and optimization decisions.
     - Data augmentation techniques.

   - Loss functions and performance evaluation metrics used.
3. **Architectures visualization:** The content pages must include visualizations of the proposed models. The architecture visualizations must be diagrams; **They cannot be a table or a computational graph**. The visualizations must be self-contained; to that end, you should add as much information as possible. You can (partially) rely on the captions for that.
4. **Results:** A description giving insight into the results obtained. Furthermore, the report must at least include the following:
   - A table with the experiment results, including results on the train and test sets, the different approaches, and settings.
   - A table or figure for error analysis illustrating how the models perform per accent and gender.
   - A figure depicting the impact of chosen approaches and regularization techniques on the training.

   For all figures and tables, ensure that captions are relevant and informative.
5. **Conclusions:** The report should contain a concise section that provides clear insights and conclusions that logically derive from the results **in bullet points**. This requires valid experimentation and results.

Your report must be a PDF document with a title page, and two content pages. Optionally, additional pages for references and appendices can be added. Note that the content pages need to be self-contained, and the appendices should only contain auxiliary material. If any of the above points are in the appendix, they will be considered absent.

2.1.1 Code

Your code should be written in Python and using PyTorch. You do not need to include any data or the weights of your trained model, only the code. Furthermore, your code must be based on the course material, and you should not use code or pre-trained models from third parties. This last point refers to the architecture and model training. For any pre-processing you want to perform, you can use any library that suits your needs, though torch-related libraries and packages are preferred.

The grading of the code will be based on the following points:

1. **Code Submission:** A file with the code should be submitted. The code should be well-documented and easy to understand.
2. **Code Basis:** The code should be based on the course material and should not use pre-trained models (the aim is to learn and understand the process of building and training your own models).
3. **Solution Coherence:** The code should be coherent with the solution proposed in the report. It should effectively implement the proposed architecture and experiments.
4. **Experiments:** The code should show the training and evaluation of different approaches and settings as described in the results in the report.
5. **Readability and comments:** The code should be organized, easy to read, and with enough explanations in the form of comments and doc strings.

6. **Competition Submission:** The results on the test set should have been submitted on the competition website. This will allow for an unbiased evaluation of the model's performance. The results on the test set must be reported.

Finally, you can base your solution on the literature (e.g., scientific articles) as long as you implement it fully without using existing code or pre-trained models.

2.1.2 Submission format

Your report and code should be in a single zip file named with your group ID (e.g., $group_1.zip$) and submitted through the assignment on Canvas by May 23rd, 2025.

2.2 Performance

In addition to the report, you must submit your results to a competition. Your submission should be under your group name (e.g., Group 1). **The accounts used should be created with a TiU email; otherwise, it won't be accepted to the competition**. Also, if the name used for the submission does not match the given format, it will be ignored. A separate document about submission to the competition will be provided with additional details later.