Introduction to Deep Learning: Practical Assignment

Due by: May 26th, at 17:00

Description

Hello, Hallo, Hola, Olá et Bonjour!

As you may have guessed by now, we will look at **spoken language detection** in this assignment. Language detection is a crucial step in machine translation. In this project, we will be training a deep learning model to accurately classify audio clips of German, English, Spanish, French, Dutch, and Portuguese.

We will provide you with a dataset of a thousand audio clips per language. Your task will be to **classify an audio clip's language using deep learning techniques**.

To guide you through the project, we have prepared **six graded questions** to help you. **Your report must be a one-page written summary of your approach and results, accompanied by the Python code used to train the model and process the data.** To complete this project, you must use PyTorch and create the models from scratch using available PyTorch layers. Pre-trained models are not permitted.

Attached to this assignment is a **grading matrix** that we will use to determine the grade. Keep in mind that **we will score you on a variety of metrics**, of which the performance of your deep-learning solution is but a small part.

The dataset can be found at: DATASET_URL. A skeleton code to get you started with the assignment is provided in canvas. **You must follow the structure laid out in the skeleton code**.

After the practical information section of this document, you will find questions that can guide you through the process of addressing this assignment. The grading will not be based on the questions, but they are related to the sections of the report you have to deliver.

Practical information

This is a group assignment. Groups should be of 3 to 4 people. For this assignment, you must implement and evaluate a model on a given test set. You must submit a report describing your solution(s) and the code you used and evaluate your model on a test set. Your model's performance will be compared to the one of a baseline model, but note that the grading is not based on the performance but on the report and code. However, a good performance on the test set can grant you bonus points.

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

Passing the assignment is not mandatory to pass the course, but it is highly advisable to hand in your solutions as not doing so implies getting a 0 on 30% of your final grade. There will be no resit for this assignment as passing it is not compulsory, and the course can be passed without passing the assignment. However, the exam may include questions that might be easier to answer if you have worked on the shared task.

Report

A document should be submitted by May 26th, at 17:00. It should include a one-page report (content page), a page specifying the role of each group member in the assignment, and, if considered adequate, an appendix page.

Content page

The content page should be self-contained and include the following sections:

- Data loading and processing
- Architecture design, where you describe your best architecture and include a diagram of the architecture as an image.
- Experiments: Briefly describe your experiments, including training, hyperparameter tuning, and optimization.
- Results: A general description of your results, a table comparing the results for different settings or architectures, and an error analysis around a graph or table.
- Short discussion and conclusions on the performance of your solution in bullet points.

Work distribution

Your report needs to contain a detailed description of who did what, so make sure to keep track of this information.

Note: it is unacceptable just to state that "All members worked together and contributed equally".

If there are any problems with collaboration, such as serious disagreements, a group member not contributing, or a group dissolving, make sure to inform the course's lecturers as soon as possible via email.

In cases of a serious imbalance in the contribution of group members, the assignment grade for each member will be adjusted to reflect that.

Appendix

Optionally, you can have an additional page for references and appendices. Note that the content page needs to be self-contained, and the appendices should only contain auxiliary material. If any of the points listed above for the content page are in the appendix, they will be considered absent.

Format

Report

Your report should be a PDF document with a single page of content, a page describing the work distribution and, optionally, an appendix page.

Code

Your code should be a plain Python script ('.py', not a notebook) that can be run to generate your predictions. You do not need to include the training data or the weights of your trained model in your submission in Canvas. You must use PyTorch for this assignment.

Canvas submission

For the submission, 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.

Performance and competition

In addition to the report, you must submit a file with the predictions to a competition website. You will need an for the group. There will be a separate document about the submission to the competition.

You can get a bonus on the assignment's grade based on your ranking on the competition's leaderboard. Specifically:

- if your rank first, you will receive a bonus of 2 points;
- if your score is no better than a provided baseline, you will receive no bonus;
- for intermediate ranks, the bonus points will be linearly interpolated. The performance of the baseline solution will be shown on the leaderboard.

All the information about the competition will be specified in a separate document later on.

Communication

For clarifications (not answers) please post in the **canvas discussions page**, or ask your questions during the lectures.

Questions

Here you can find questions that can guide you while working on your assignment. Note that the grading will not be based on these questions, but they are related to the sections of the report you need to write.

Question 1: Data loading and processing

The dataset consists of individual waveforms. These contain the shape of the sound signal over time. Because our audio clips are digital, the waveforms consist of sound amplitude at invidual timesteps. The clips here are sampled at 8 kilohertz (khz), meaning there are 8000 amplitude measurements for every second. Each audio clip is 5 seconds long. Each audio clip has $8000 \cdot 5 = 40000$ measurements in total.

Question 1a: With this in mind, plot the waveforms of 5 randomly selected samples from the training set. Make sure that for each image, the y-axis has the same limits. The resulting image should look something like this:

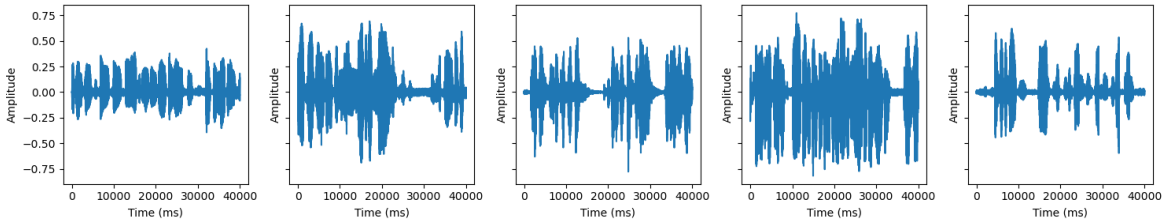

Figure 1: The waveforms of 5 arbitrary audio clips.

Apart from the number/pattern of peaks, there is a marked difference between the plots. This difference has nothing to do with the language being spoken. Explain the difference in one sentence. In addition, explain what normalization step can be taken to prevent a model from overfitting on this difference in two sentences or less.

Question 1b: Perform normalization and transformation of the input data. In tutorials, we have seen a basic way to perform data normalization. We can take this step. In addition, take the normalization step you discussed in **Question 1a.**

Question 2: Training a binary classifier

Having processed the data, we can move on to the fun stuff! We will train a neural network to distinguish between English and Spanish language. Select all dataset samples where the target is either 1 (English) or 2 (Spanish). To all English samples, assign the target 0, and to all Spanish samples, assign the target 1.

Question 2a: Construct a neural network that takes in the samples and outputs a probability of a clip being Spanish vs English. Remember that the network must be able to map very long sequences to a single output without millions of parameters. Defend the design choices of your network (i.e., type of layers, number of layers, order of layers) for the task at hand **within 150 words. You do not need a model with millions of parameters to obtain good performance! A 100.000-parameter model can perform adequately if correctly designed.**

Question 2b: Given that this is a binary classification problem, what loss function could be best used?.
Question 3b: Write a training loop and train the model for a few epochs (5 to 20).. Use the Adam optimizer. Choose a suitable learning rate (A good place to start may be 10^{-3} to 10^{-4}).

Save the loss on the training and validation test set every 10 steps. After training, plot the results. The resulting Figure could look something like this:

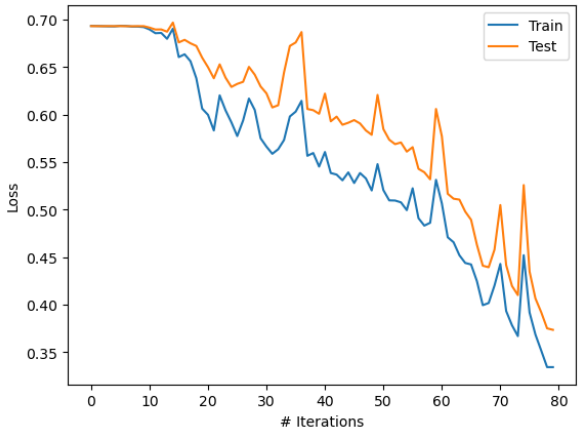

Figure 2: Development of training/test loss

What can you say about the difference between training and validation set loss in your Figure? With which techniques can this difference be made smaller? Apply one or more of these techniques and explain in two sentences what the result is. If the model has both low training and test loss, explain why you believe there is no apparent difference already. (E.g., you already applied one or more techniques).

If you have successfully trained a model to 80%+ test accuracy on this task, you are on the right track! No large changes to your current model will be necessary from here on out.

Question 3: Training on all six languages

Now, we will attempt to distinguish all six languages from each other using deep learning.

Question 3a: Modify your model to support the six classes. Thus, at least the output layer should be changed. Remember that this task is more difficult than in **Question 2**. You may need to increase the network's capacity slightly to achieve similar performance (e.g. adding layers or increasing the number of neurons within the layers). In addition, consider the role of batch size and learning rate.
Question 3b: What loss function is best suited for this task?.
Question 3c: Train the model for a few epochs (10 to 20). Provide a plot of a confusion matrix of the model on the test set.

What do you observe? Is there something relevant that you can explain (In 100 words or less)?

Question 3d: Analyse the test set in your model's output space using dimensionality reduction. For this, we will be using a technique called PCA (principal component analysis). Follow the instructions in the skeleton code; the picture could look something like this:

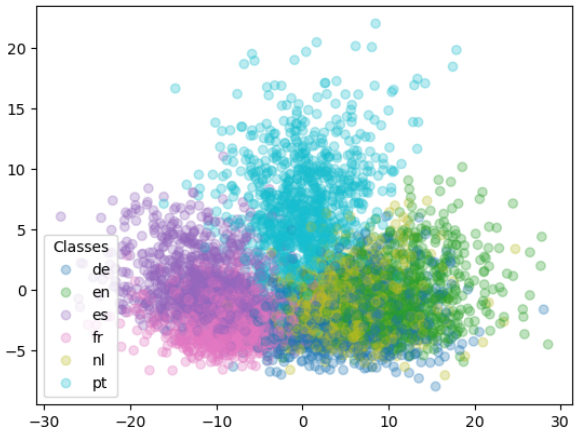

Figure 3: PCA on the output space of a trained classifier.

Does the picture make 'sense' to you? For example, are the distances between the languages within the model's output space logically explainable? Explain also how this relates to the answer of **Question 3c**, in 200 words or less.

Question 4: Bias in the data

The dataset your model is now trained on was mined from YouTube videos using language-specific searches.

Question 4a: Given this knowledge, when could the model deliver biased results when applied to general use cases?

Bonus question

Does your model work on sequences of arbitrary length? Or do you have to trim the clip to 5 seconds first? If your model works on series of arbitrary length, explain why in less than 100 words. If not, explain how you might change the model to achieve this in 100 words or less.

Good luck!
¡Buena suerte!
Viel Glück!
Veel succes!
Bonne chance!
Boa sorte!

Cascha, Sebastian and Dimitar,