Report Group 37

Aryan V Verma (2060326), Benjamin Matthews (2061562),
Dominic Stamatoiu (2051167), Osama Al Hassan (2048767)

Data loading and processing

The data consisted of 1000 samples of 5 second audio for 6 different languages (French, Spanish, English, Dutch, German and Portuguese). We found the dataset to be quite limited so we used 6 methods of 4 data augmentation techniques (Guassian noise injection, Pitch shift, Normal shift, Time stretch, A combination of all 4 and A random selection of 2) from the audiomentations library to increase the size by 700%, leaving the cumulative total of the training data at 42,000 samples.

Architecture design

The architecture in Figure 1 was found to be the best-performing one, when experimenting by combining different layers, pools and functions. When looking at the deep learning algorithm that should be used, Convolutional Neural Networks came to our mind as they are well suited for audio classification and for extracting spatial and temporal features from spectrogram representations of waveforms. Apart from using Relu for its non-linearity purposes, we considered that Softmax would be a good choice for the activation function as it works well with multi-classes, in our case six, and gives us the language name with the highest probability of being presented in the audio. Adding the adaptive average pool on top of the average pool as well as adding multiple hidden layers in the network further improved the results.

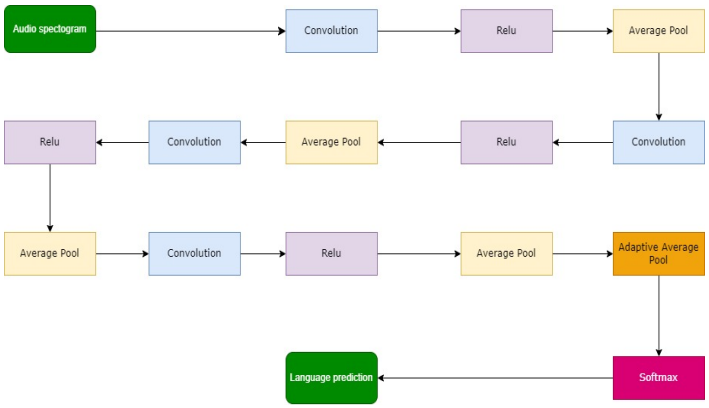

Figure 1: Diagram summarizing the proposed architecture.

Experiments

We built two models based on different architectures as well as running multiple experiments on the displayed model using the Adam (Adaptive Moment Estimation) optimizer due to its ability to handle noisy and sparse data ((Mahendra, 2023)). The learning rate and the epochs were manipulated, as shown in Table 1. We also switched loss functions between NLL (Negative Log Likelihood) and CE (Cross Entropy) to compare performance. Learning rate, weight decay and epoch manipulations were kept consistent over both loss functions.

Results

As shown in the results in Table 1, NLL with a learning rate of 0.015 and 25 epochs was the best performing, while CE with the same learning rate and 40 epochs was the worst. We suspected the higher number of epochs led to overfitting , which led to worse performance. Cross entropy was also the significantly worse loss function in our trials. The heatmap presented in 2 showcases that the most commonly misclassified language was English which was predicted as Dutch 28 times.

Loss Func	LR	WD	Epochs	Accuracy
Neg Log Likelihood	0.015	0.0001	25	82%
Neg Log Likelihood	0.01	0.0001	25	81%
Neg Log Likelihood	0.015	0.0001	40	79%
Cross Entropy	0.015	0.0001	25	74%
Cross Entropy	0.01	0.0001	25	72%
Cross Entropy	0.015	0.0001	40	68%

Table 1: Table presenting the results

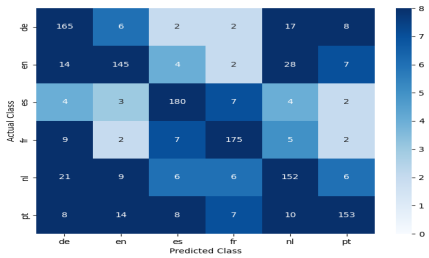

Figure 2: Heatmap of missclassifications.

Conclusions

- We concluded that the architecture provided in this report (first iteration) was deemed successful as it achieved max accuracy and Negative Log Likelihood served as the better loss function in this use case, as also stated by (Yao, Zhu, Jiang, & Yu, 2019) for image classification
- We also concluded that the number of epochs in the first iteration was the max to which the accuracy could have been increased. Further adding epochs only decreased it.
- The main classification problem of the audio was between Dutch and English or German, for which we reasoned that this is due to the high linguistic similarity between the languages (Gooskens et al., 2018).
- An Lr of 0.015 provided faster training and better-optimization of the final weights in comparison to 0.01 which can be attributed to large learning rates allowing models to learn faster, according to (Perlato, 2020).

Work Distribution

For this project, all members aided in investigating different architechtures. Aryan and Dom were responsible for A CNN model with 3 convulational and 3 linear layers whereas Ben and Osama were responsible for investigating A CNN model with 4 sets containing a convolutional layer, Relu with batch normalization and pooling in each set. On deciding the final architecture, Ben and Osama continued working on the model as they were familiar with it and carried out the hyperparameter tuning. For the report itself, Ben was responsible for the data loading section, Dom created the diagram and architecture design section along with the conclusion, Osama did the experiments section and Aryan worked on the results and conclusion.

References

Gooskens, C., Van Heuven, V. J., Golubovic, J., Schüppert, A., Swarte, F., & Voigt, S. (2018, 4). Mutual intelligibility between closely related languages in Europe. *International Journal of Multilingualism*, *15*(2), 169–193. Retrieved from https://doi.org/10.1080/14790718.2017.1350185 doi: 10.1080/14790718.2017.1350185

Mahendra, S. (2023, 3). What is the Adam Optimizer and How is It Used in Machine Learning. *Artificial Intelligence +*. Retrieved from https://www.aiplusinfo.com/blog/what-is-the-adam-optimizer-and-how-is-it-used-in-machine-learning/

Perlato. (2020). *The Learning Rate - Andrea Perlato.* Retrieved from https://www.andreaperlato.com/theorypost/the-learning-rate/#:~:text=Generally%2C%20a%20large%20learning%20rate,take%20significantly%20longer%20to%20train.

Yao, H., Zhu, D., Jiang, B., & Yu, P. (2019). *Negative Log Likelihood Ratio Loss for Deep Neural Network Classification.* Springer Nature. Retrieved from https://doi.org/10.1007/978-3-030-32520-6_22 doi: 10.1007/978-3-030-32520-6\{_}22