



WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF MATHEMATICS AND INFORMATION SCIENCE

Project II Report

Deep Learning

Students:

Karina Tiurina (335943)

Mateusz Zacharecki (313549)

Supervisor:

mgr inż. Maciej Żelaszczyk

Warsaw 2024

Contents

1 Research problem.....	3
2 Application instruction	4
3 Theoretical introduction	5
4 Conducted experiments	7
4.1 Transformer	7
4.2 ConvS2S.....	10
4.3 LSTM/GRU	12
5 Summary	14
References.....	15

1 Research problem

The topic of the project is Audio Classification based on the data from 'TensorFlow Speech Recognition Challenge' on Kaggle:

<https://www.kaggle.com/competitions/tensorflow-speech-recognition-challenge>.

The dataset contains one second audio files which should be classified into 12 classes: 'yes', 'no', 'up', 'down', 'left', 'right', 'on', 'off', 'stop', 'go', 'silence' and 'unknown'.

The aim of this project is to train and compare different classification architectures, including a Transformer architecture.

2 Application instruction

Source code has the following structure.

- convs2s
 - o test.ipynb
- transformer
 - o transformer_lstm_gru.ipynb
 - o data-preparation.ipynb
- kaggle_submissions
 - o 0.81099_wav2vec2-base.csv
 - o 0.85727_AST.csv

To reproduce the results presented in this report, please, run /convs2s/test.ipynb and /transformer.ipynb. For LSTM and GRU models, please, run data-preparation.ipynb first.

To prepare submission files and confusion matrices, the path to the models should be updated accordingly.

3 Theoretical introduction

The Transformer model, introduced in 2017, is a neural network architecture primarily used for natural language processing tasks. Unlike traditional models, it relies solely on self-attention mechanisms, allowing it to capture relationships between words in a sequence more efficiently. This architecture, featuring multi-head attention and feed-forward layers, has become a cornerstone in NLP due to its ability to handle long-range dependencies and achieve state-of-the-art results across various language tasks.

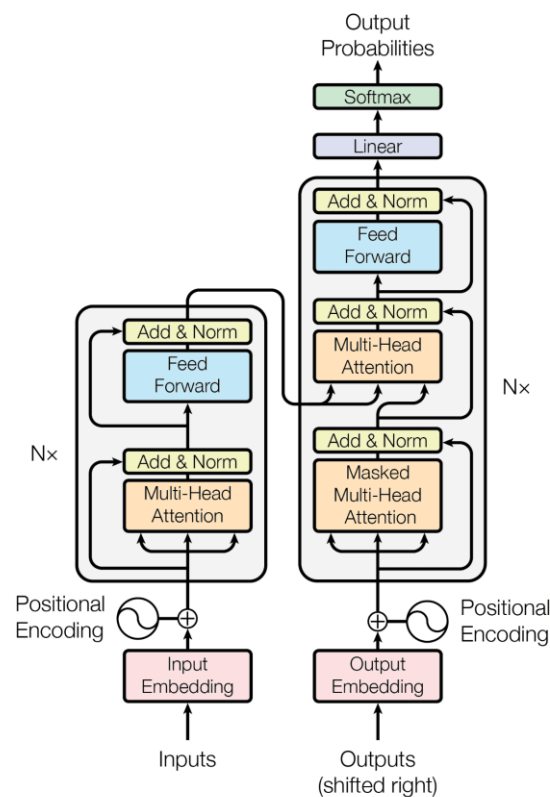


Figure 3.1 Transformer architecture

The Convolutional Sequence to Sequence model (ConvS2S), introduced in 2017, is an architecture which is entirely based on convolutional neural networks (CNN). Computations can be fully parallelized during training. Gated linear units (GLU) eases gradient propagation and each decoder layer equipped with a separate attention module.

- At the top part, it is the encoder. At the bottom part, it is the decoder.

- The encoder RNN processes an input sequence $x=(x_1, \dots, x_m)$ of m elements and returns state representations $z=(z_1, \dots, z_m)$.
- The decoder RNN takes z and generates the output sequence $y=(y_1, \dots, y_n)$ left to right, one element at a time.

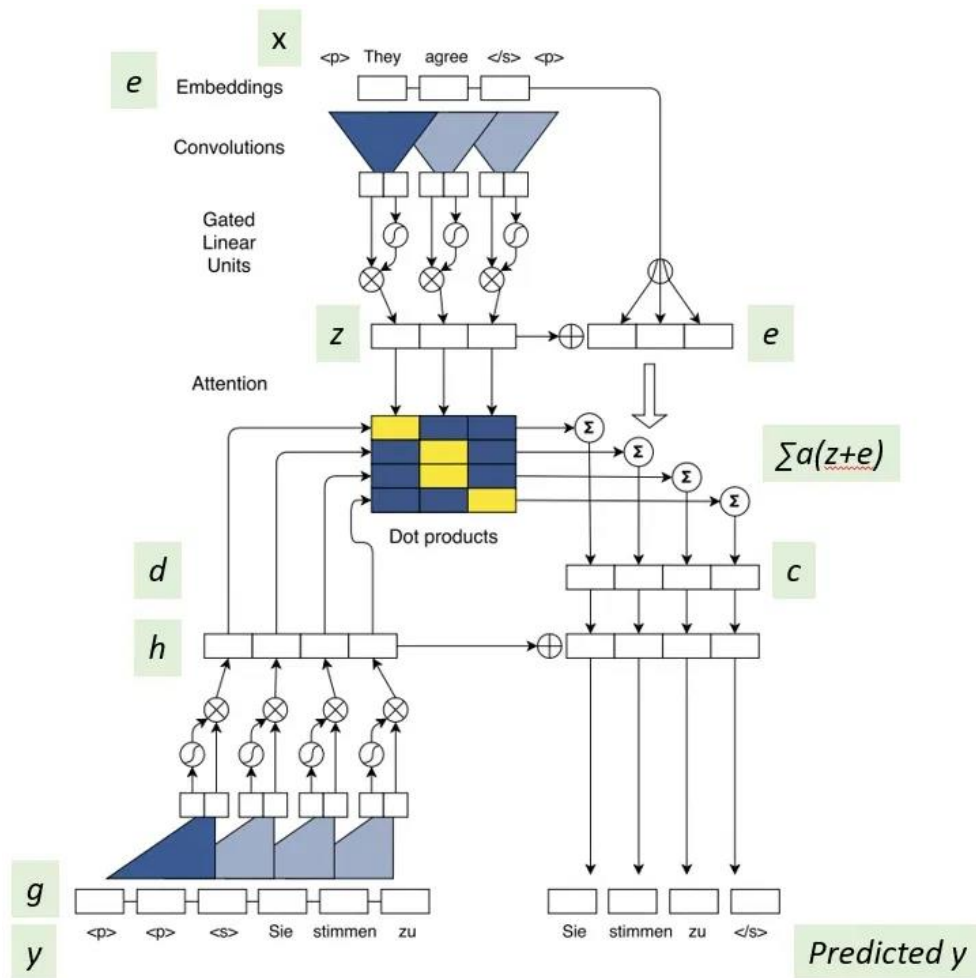


Figure 3.2 Convolutional Sequence to Sequence (ConvS2S) architecture

4 Conducted experiments

4.1 Transformer

To train transformer model, a python package 'transformers' was used with pretrained models available on Hugging Face.

The first model was trained to classify 30 speech commands. If the score for the best class is lower than 0.5, the model predicted 'unknown' class. If score was lower than 0.3 - 'silence' class. The accuracy on the validation dataset was 97.95% with the confusion matrix presented on figure 4.1.

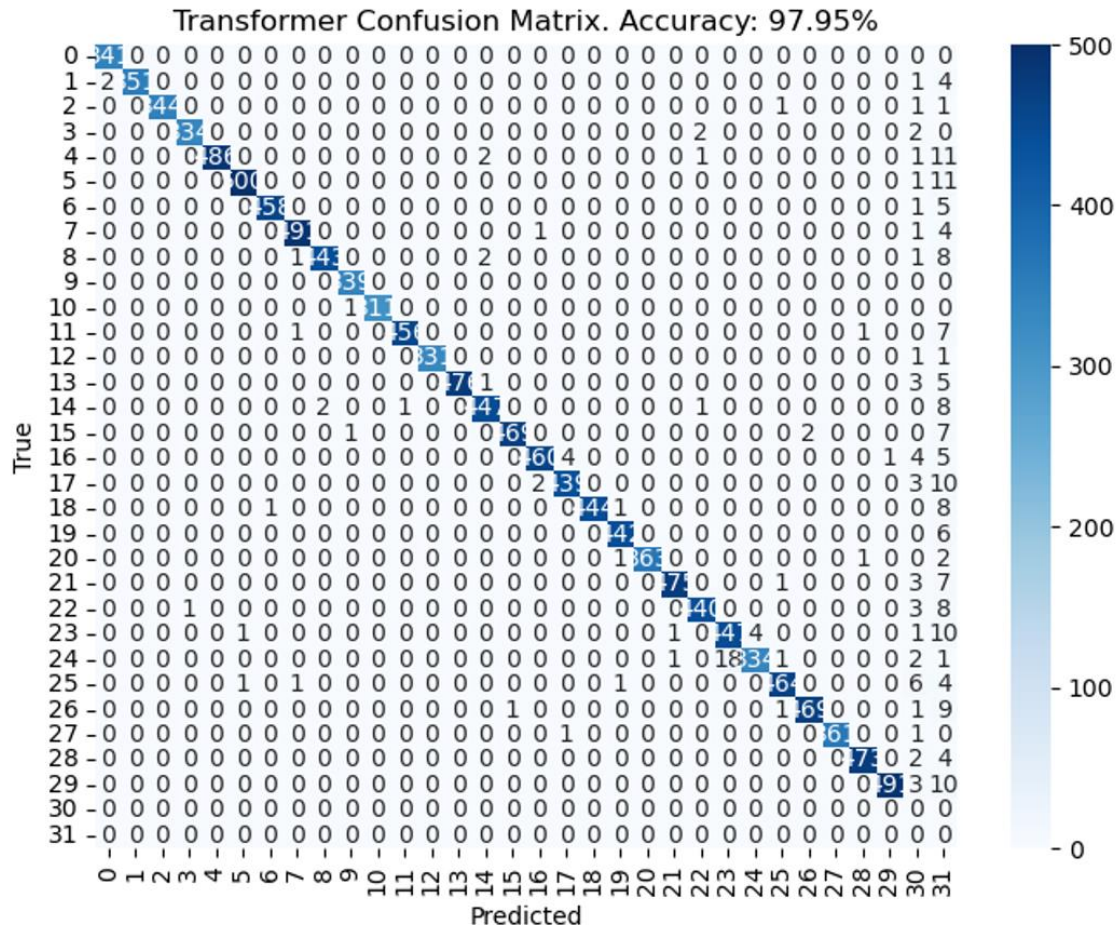


Figure 4.1 Confusion matrix for transformer wav2vec2-base

On Kaggle this model provided 81.09% accuracy on the private dataset.


Submission and Description		Private Score ⓘ	Public Score ⓘ
	0_submission.csv	0.81099	0.80016
	Complete (after deadline) · 1d ago · Transformer (facebook/wav2vec2-base)		

Figure 4.2 Kaggle submission of transformer wav2vec2-base

Then the number of classes was reduced to 10 and all other known classes relabeled to 'unknown'. 'silence' class was augmented from existing 'background noise' audio. The initial accuracy was around 60%, however, all of the validation data was predicted as 'unknown'.

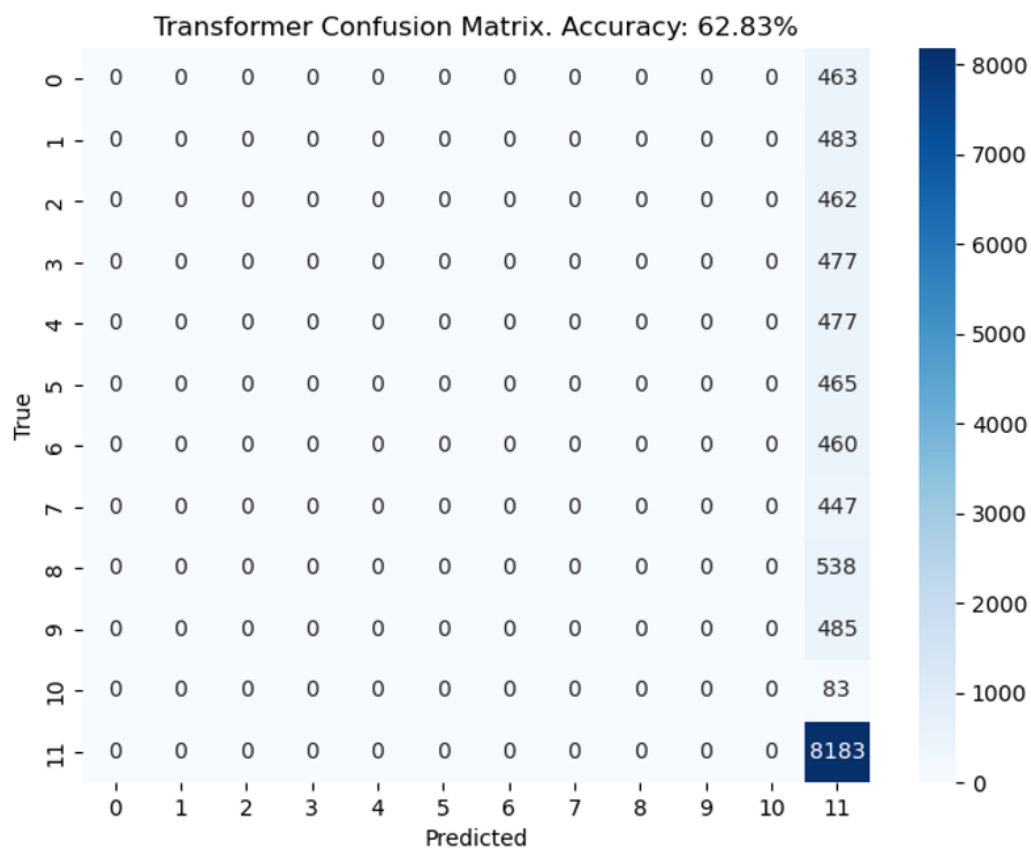


Figure 4.3 Confusion Matrix on relabeled data

Overestimation on 'unknown' label was due to the fact that after the relabeling of classes 60% of the data became 'unknown'. Reduction of the amount of 'unknown' examples to ~2000 improved the model performance back to 97.32%.

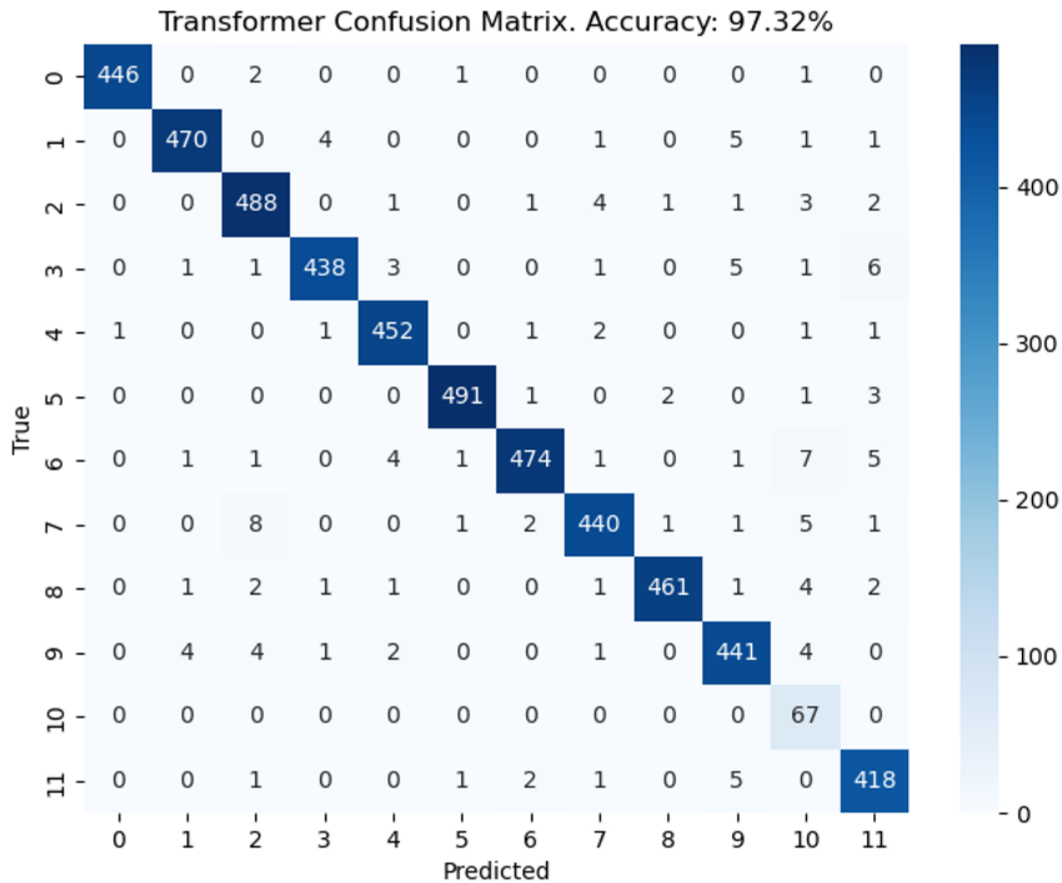


Figure 4.4 Confusion matrix with relabeled data and reduced 'unknown' amount

Another test was conducted on checking a different pretrained model available on Hugging Face: Audio Spectrogram Transformer (AST) MIT/ast-finetuned-speech-commands-v2. This model was trained to specifically classify speech commands, which might improve the performance, especially on Kaggle.

The accuracy on the validation set has slightly improved to 97.7%.

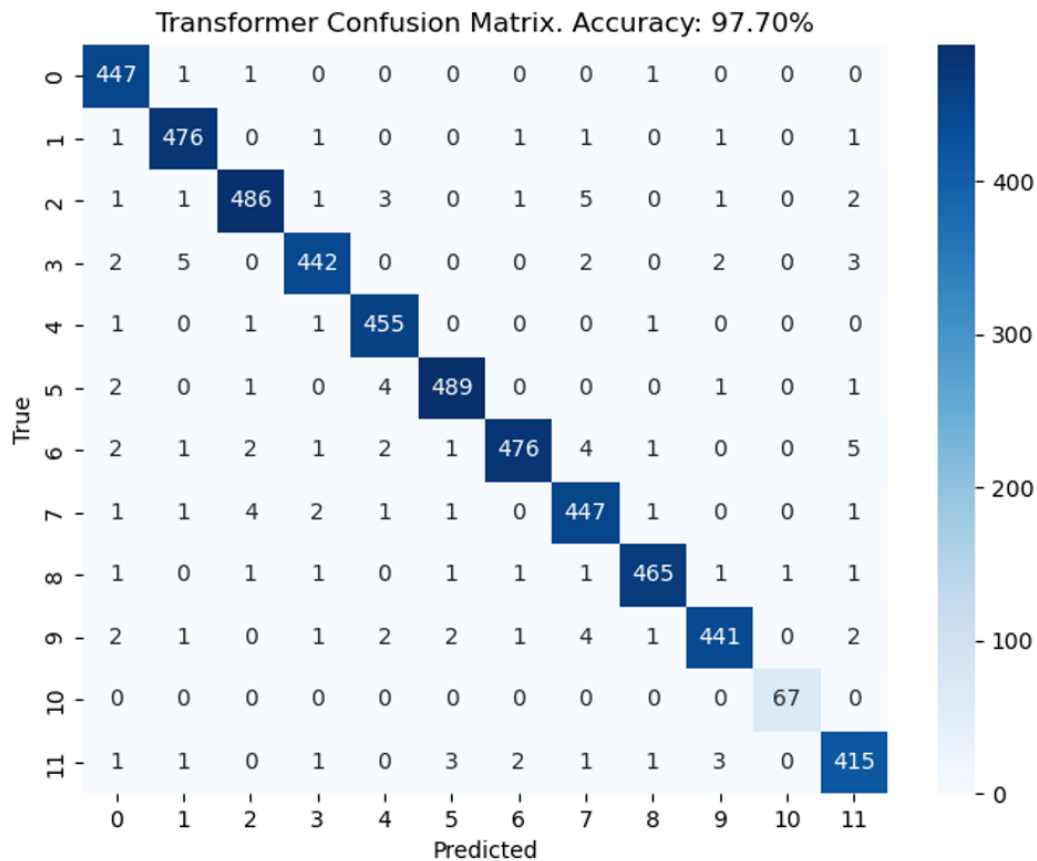


Figure 4.5 Confusion matrix of AST transfer learning

AST model on Kaggle has improved on both public and private score up to 85.72%.


Submission and Description		Private Score ⓘ	Public Score ⓘ
 0_submission.csv Complete (after deadline) · 3h ago · MIT/ast-finetuned-speech-commands-v2			
		0.85727	0.84703

Figure 4.6 Kaggle scores of AST

4.2 ConvS2S

For Convolutional Sequence to Sequence model there were 7 different tests conducted. Source code and results of the training are in /convs2s folder.

1. convs2s_1 – ConvS2S model with mse loss and adam optimizer

2. convs2s_2 - ConvS2S model with mse loss and adamax optimizer
3. convs2s_3 - ConvS2S model with mse loss and sgd optimizer
4. convs2s_4 - ConvS2S model with categorical crossentropy loss and adam optimizer
5. convs2s_5 - ConvS2S model with mse loss and RMSprop optimizer
6. convs2s_6 - ConvS2S model with mse loss and RMSprop optimizer for data with reduced unknown class observations
7. convs2s_7 - ConvS2S model with mse loss and RMSprop optimizer for data with reduced unknown class observations and for batch_size = 256

The first 5 tests were conducted for observations of not considered classes assigned as unknown class and data augmentation for silence class. The last 2 tests were conducted for additional reduction of observations from unknown class to about 2000 in order to get similar numbers of observations from each class and, as result, get better accuracy.

The best results we get for convs2s_5 model.

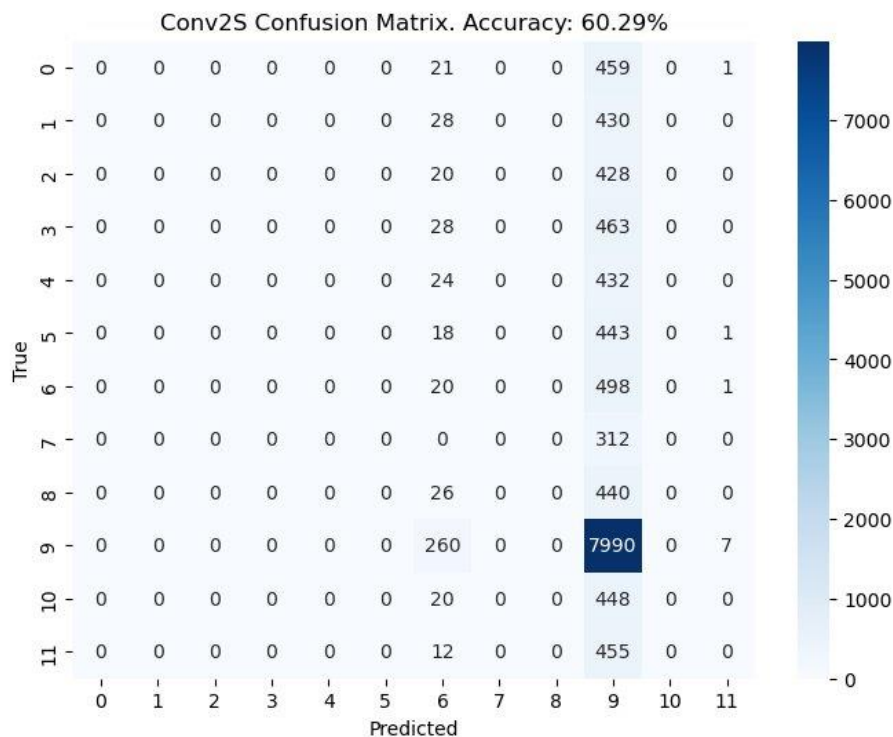


Figure 4.7 Confusion matrix for ConvS2S_5 model.

Unfortunately, we get for convs2s_6 the same predictions for every observation and as result we get such confusion matrix and accuracy.

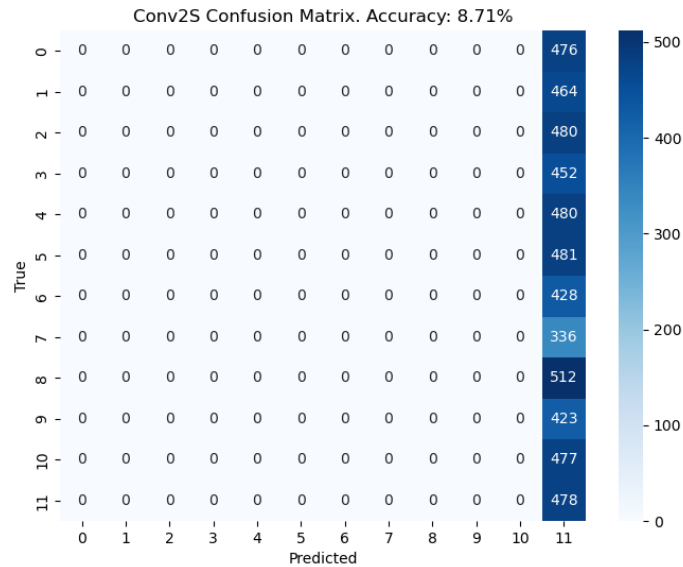


Figure 4.8 Confusion matrix for ConvS2S_6 model.

The last model is then the attempt to solve this problem. Unfortunately, setting `batch_size = 256` does not improve results and we get similar confusion matrix.

4.3 LSTM/GRU

Unfortunately, LSTM and GRU model can be considered a failures in our case. They were tested based on preprocessed data: melspectrogram created from the amplitude of the sound. The accuracy provided was only about 3% of accuracy on the validation set which can be considered a complete random choice between 30 classes.

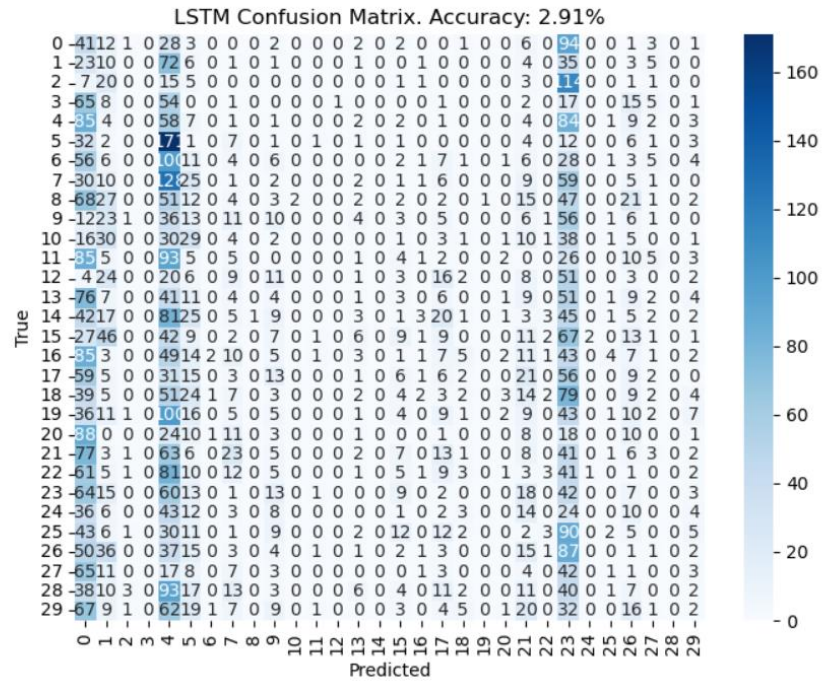
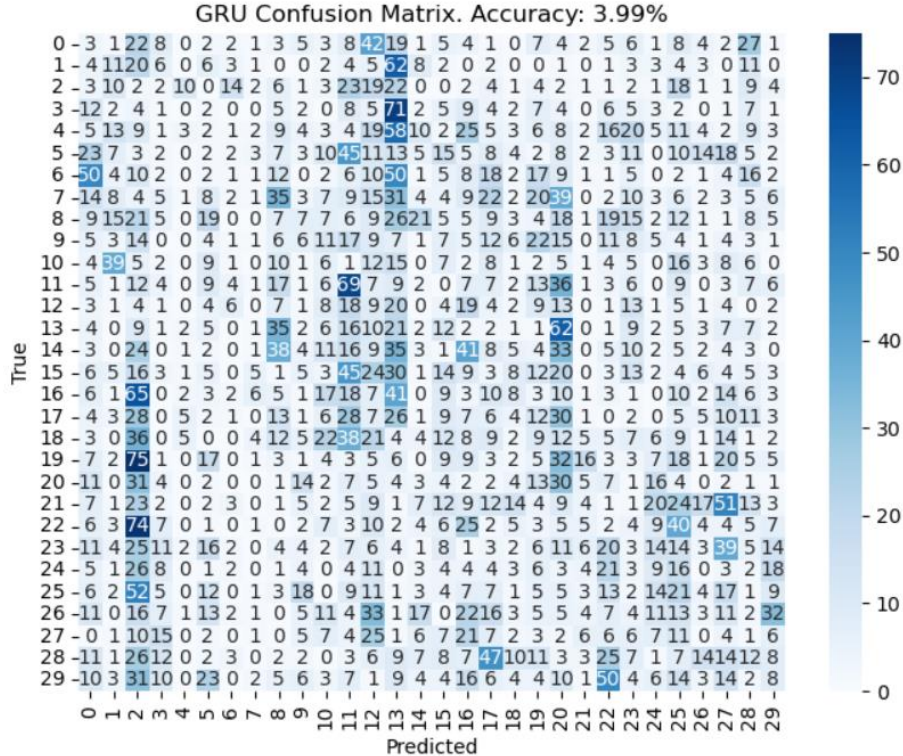


Figure 4.9 LSTM confusion matrix



5 Summary

Crucial for good results was surely the appropriate preparation of data. As treatment all observations with classes that weren't considered in this classification task as unknown class resulted in assigning most of observations on validation set as unknown, reduction of them let classify observations from other classes much more correctly. Also training in all 30 classes didn't give satisfying enough results, especially on 'silence' and 'unknown' labels. For appropriate silence class classification, data augmentation was crucial. Additionally, failure in LSTM and GRU testing might be because of a wrong data preprocessing or an incorrect choice of the network architecture.

The following research could be further conducted to improve learning accuracy.

1. Test more hyperparameters, both related to training process and related to regularization.
2. Select and test subsets of classes.
3. Train separate network for silence and unknown classes recognition.
4. Consider other pre-trained models, e.g. DeepSpeech2.
5. Test different network architectures for LSTM and GRU models.

References

- [1] Tensorflow Speech Recognition Challenge
<https://www.kaggle.com/competitions/tensorflow-speech-recognition-challenge>
- [2] <https://sh-tsang.medium.com/review-convolutional-sequence-to-sequence-learning-convs2s-510a9eddce05>
- [3] <https://www.kaggle.com/code/araspirbadian/voice-command-detection>
- [4] <https://www.kaggle.com/code/davids1992/speech-representation-and-data-exploration>
- [5] Attention is All you Need, *Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin*
- [6] <https://nlp.seas.harvard.edu/2018/04/03/attention.html>
- [7] Hugging Face <https://huggingface.co/>