Tutorial 05: CNNs for Audio, Regularisation

The goal of this exercise is to run Deep Learning Experiments on audio data. Audio is a good example to learn about the importance of feature extraction for machine learning, as the raw data of an audio signal (e.g. 48 000 1-Dimensional values per second) is quite difficult to handle. The exercise is therefore intended as a deep learning practice for common types of databases, which need data preprocessing. The Instructions on this sheet are often rather vague, as part of the exercise is to figure out, how to use which part of the data set to obtain results.

Please submit your code and results by 10th Dec 23:59 to manuel.milling@informatik.uni-augsburg.de. You can submit your solutions alone or in Teams of 2 (please indicate all names with the submission).

1 Urban Sound Data Set (2P)

Download the urban sound data set https://megastore.uni-augsburg.de/get/QwNq1a0XpF/ and unpack it. The data is only meant to be used for this course. For further information on the dataset see regarding license see https://urbansounddataset.weebly.com/download-urbansound.html and https://creativecommons.org/licenses/by-nc/3.0/.

2 Spectogram Extraction (7P)

Implement and run a routine that converts the individual wav files into spectograms of a fixed length (work out a good maximum length based on the files). You can use the library **librosa** to create the spectograms (**librosa.feature.melspectrogram**). Compared to the defaults values, you can vary parameters like **window_size**, **hop_length** and **n_fft** to produce different spectograms.

3 Data Loading (7P)

Implement a routine that loads the spectograms and matches them to their labels according to the meta-data file. The data should be stored in numpy arrays and roughly 10% of the data should be reserved for testing.
*Note: The library **pandas** can be useful to load and process data from csv files.*

4 Initial CNN model and training (4P)

Reuse and adapt the CNN model from the previous exercise sheet and train the model on your data set. Submit your results from the training (accuracy should be above

50%). Do you observe the effect of overfitting on the data?
Note: You can use google colab for fast training of your model. The files for this purpose need to be uploaded into your google drive.

5 Bonus: Model optimisation (6P)

Try to increase the test results. Try different network architectures, different spectograms and/or regularisation and submit your results. Try to beat 55% (2P), 60% (2P) and 65% (2P) test accuracy.