Report

Group 45

Group members:

- Jan Kaleta-Słyk (2075701)
- Tomasz Kubrak (2082917)
- Rafał Kukiełka (2060041)
- Tae Yong Kwon (2082154)

Data loading and processing

We defined a function to load and immediately preprocess the data. First, we normalized it by specifying the target length of the utterances to 61500. This value represents the maximal length that 90% of the utterances were equal to or shorter than. If they were shorter than the target, they would be padded with “0”s, whereas if they were longer than the target, they would be trimmed to match it. Next, they were converted from NumPy to PyTorch tensor. Normalization of the data was necessary to use it as an input for a CNN, as it expects fixed-size tensors. Secondly, we created the Mel spectrogram using PyTorch’s Mel Spectrogram class and set the following parameters: sample_rate = 8000, n_mels = 64, hop_length = 512, n_fft =1024. We used the Mel spectrograms to reduce the input dimensionality by preserving the most important features for the prediction of valence.

Architecture design

The network consists of four blocks, each containing a convolutional layer with a kernel size of 3x3, stride of 1, and padding of 7, followed by a ReLU activation function and an average pooling with a kernel size of 2. We used the ReLU activation function to introduce non-linearity to the model, allowing it to learn more complex patterns. Average pooling was used to reduce the dimensionality of each feature map, helping to prevent the model from overfitting. This architecture progressively reduces the spatial size of the representation, increasing the depth (number of channels) while capturing higher-level features at each layer. After the convolutional and pooling operations, a flatten layer was used to convert the output to a one-dimensional tensor. This transformation is necessary for connecting the convolutional layers to the fully connected layer. The fully connected layer maps the flattened features to a single output value, combining the learned features to produce the final prediction - the valence of each sample.

Experiments

To find the optimal hyperparameters, a grid search was conducted with the following parameters: learning rate (0.001, 0.0001, 0.0005, 0.00005, 0.00001), activation function (ReLU, Sigmoid), kernel size (3x3, 5x5), and pooling method (max, average). Each combination was tested using 3-fold cross-validation on a validation set. Testing different learning rates aimed to balance convergence speed and stability. Higher rates (0.001) could lead to faster convergence but risk overshooting minima, while lower rates (0.0001) might ensure stable training but take longer (Sellat, 2022). Additional rates were included for fine-tuning. ReLU was initially chosen for its effectiveness in deep networks, but Sigmoid was also tested for its potential to capture fine-grained audio data details (Dubey et al., 2022). Additionally, 5x5 kernels were evaluated against 3x3 kernels to see if capturing a larger context improved performance, as larger kernels can identify more complex features (Ganjdanesh et al., 2023). Max pooling was compared with average pooling to see if a different strategy might yield better generalization, particularly in smoothing the audio signal for emotion valence prediction (Zafar et al., 2022). Each model was trained for 50 epochs using the Adam optimizer. Mean squared(MSE) error was used as the loss function to measure the difference between predicted and actual valence values.

Results

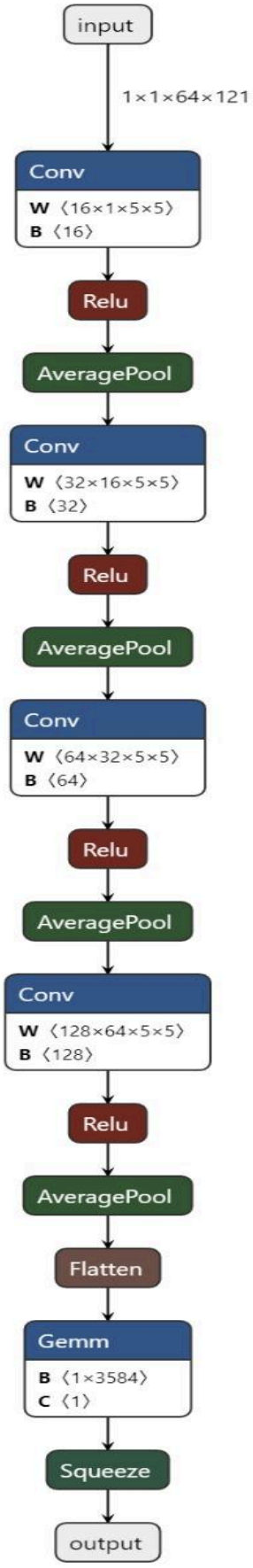
Our model achieved the lowest MSE after training for 50 epochs, with a learning rate of 0.0001, a batch size of 4, using the ReLU activation function, average pooling, and a kernel size of 5x5. These hyperparameter settings allowed the proposed model to attain MSE = 0.4665 on our validation set, and MSE = 0.48261 on the Kaggle test set.

Conclusions

- We determined that the proposed model achieves its best performance using the specified hyperparameter configuration.
- Our experiment demonstrates that even a basic CNN can, to some extent, assess the emotional valence of spoken utterances, although this design was not sufficient to outperform the provided baseline model.
- Performance of the model could be potentially improved by introducing more advanced data pre-processing and extending designed architecture by additional blocks. This modification might enhance the network’s ability to learn more complex patterns and elevate its ability to estimate emotional valence

Proposed architecture

Figure 1. Diagram of the CNN



References

Sellat, Q. (2022). Cognitive Big Data Intelligence with a metaheuristic approach. Academic press.

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in Deep learning: A comprehensive survey and benchmark. Neurocomputing, 503, 92–108. https://doi.org/10.1016/j.neucom.2022.06.111

Zafar, A., Aamir, M., Mohd Nawi, N., Arshad, A., Riaz, S., Alruban, A., Dutta, A. K., & Almotairi, S. (2022). A comparison of pooling methods for Convolutional Neural Networks. Applied Sciences, 12(17), 8643. https://doi.org/10.3390/app12178643

Ganjdanesh, A., Gao, S., & Huang, H. (2023). EffConv: Efficient learning of kernel sizes for convolution layers of CNNs. Proceedings of the AAAI Conference on Artificial Intelligence, 37(6), 7604–7612. https://doi.org/10.1609/aaai.v37i6.25923