COMSM0045 Applied Deep Learning Report

Alex Elwood
University of Bristol
wh20899@bristol.ac.uk

Artur Varosyan
University of Bristol
rm20544@bristol.ac.uk

I. Signed Agreement

We agree that all members have contributed to this project (both code and report) in an approximately equal manner.

[illegible] [illegible]

Fig. 1: Signed: Alex Elwood, Artur Varosyan

II. Introduction

Music Information Retrieval (MIR) refers to taking audio data and extracting semantic information such as genre, mood and artist. Early machine learning models relied on extracting features from raw audio signals and using methods such as support vector machines to classify the samples. The process of extracting the relevant features from the audio samples requires prior knowledge of sound engineering. More recent approaches to this problem make use of feature learning or representation learning where the model is allowed to learn the relevant features itself. Advancements in machine learning research and a higher computing capacity have allowed the research community to develop multi-layer deep neural network models, capable of accurately identifying the relevant features in a dataset.

Dieleman and Schrauwen [1] demonstrated that a multi-layer Convolutional Neural Network (CNN) is capable of identifying relevant feature representations from raw audio and making accurate predictions on an audio tagging task. The researchers used the MagnaTagATune dataset to train and analyse the performance of their model.

In this paper, we replicate the original model proposed by Dieleman and Schrauwen and train it using the same MagnaTagATune dataset. Section III describes the related work on the subject area and the inspiration behind our improved model design. Section IV describes the format and structure of the dataset. Sections V and VI describe in-depth our approach to replicating the results from the above paper. In Sections VII, VIII and IX we empirically analyse the performance of our trained model and present the results from our experiments. Finally, in Sections X and XI we present an extension to the proposed model which achieves a higher test prediction accuracy.

III. Related Work

A paper which addresses an audio tagging problem similar to the one presented by Dieleman and Schrauwen [1] is the 2016 paper by Choi et al. [2]. The researchers propose a Convolutional Recurrent Neural Network (CRNN) architecture which outperforms classic CNN architectures on the Million Song Dataset. The dataset is preprocessed to convert the raw audio files into log-amplitude mel-spectrograms, which are then fed to the model consisting of a CNN and an RNN. Choi et al. argue that CNNs perform highly on local feature extraction whereas RNNs are superior at identifying temporal patterns.

While this paper was a source of inspiration for our extension, we found the paper by Panwar et al. [3] to be more suitable. The paper builds on top of the architecture presented by Choi et al. and implements a CRNN model which performs the audio tagging task on the MagnaTagATune dataset. The new model performs highly, achieving an AUC-ROC score of 0.893. As a result, this is the paper we have chosen to replicate as part of the extension, described in greater detail in Section X.

A more recent publication by Wang et al. [4] from 2019 proposed a novel architecture labelled CRNN-TF which uses CNNs which produce activation maps of mid-level time-frequency representations. It then uses Long Short-Term Memory (LSTM) blocks within the RNN to extract the spatial dependencies. This allowed the authors to achieve a start-of-art performance on the MagnaTagATune dataset with an AUC score of 0.9078.

IV. Dataset

The MagnaTagATune dataset consists of 25,863 raw audio clips each with a vector of binary annotations representing 188 tags. The annotations include but are not limited to genres, such as *classical* and *jazz*, but also other tags such as *singer* and *violin*. As in the original paper by Dieleman and Shcrauwen [1], we use a dataset containing only the 50 most common tags. Each raw audio clip is a time series with one channel where each value is a signed 16-bit integer. The audio clips are sampled at a rate of 16kHz for 29.14 seconds.

However, there are some known drawbacks of the dataset. Firstly, the tags of the dataset are highly unbalanced (a visible issue within Figure 5), however, taking the top 50 tags mitigates this problem [5]. Additionally, Marques et al. [6] reports multiple issues, one particular example being antonymy. This means that some clips have tags with opposite meanings, for

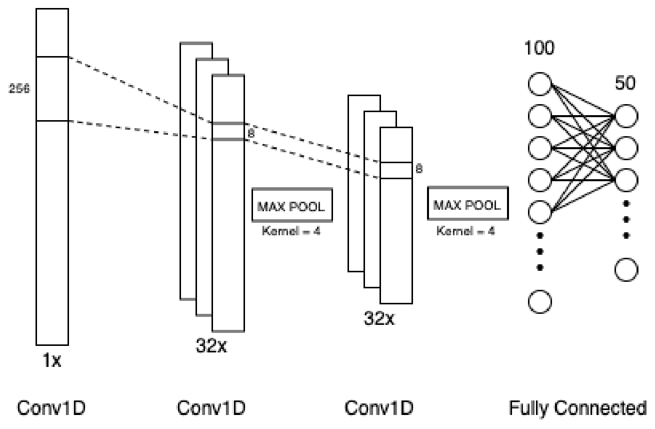

Fig. 2: Base Architecture Diagram

example, loud and quiet. These issues should be considered in the analysis of any model trained on MagnaTagATune.

V. CNN Architecture

The mono-channel one-dimensional audio signal is first passed through a convolutional layer which outputs 32 channels. The length and stride of this layer are parameterised. The second and third layers are convolutions with a fixed length of 8, stride of 1, and padding to preserve dimensionality; which produce 32 channels. The signal is then flattened and passed through a fully connected layer with an output size of 100, and a second fully connected layer with an output size of 50.

After each convolutional layer and the first fully connected layer, the signal is passed through a Rectified Linear Unit (ReLU) function. In addition, after the second and third convolutional layers and respective ReLU, max pooling with kernel size 4 is applied. The sigmoid function is then applied to the output of the final fully connected layer and the resulting vector of size 50 is the logit vector. A visual representation of the architecture is displayed in Figure 2.

VI. Implementation Details

A. Data Processing

Before training the model, we modified the dataset to separate the validation set from the training set. This involved restructuring the dataset directory and modifying the label annotations to reference these new paths. As in the original paper, we used the first 12 parts of the dataset for training, 1 for validation and 3 for testing.

Although Dieleman and Schrauwen did not mention normalisation in their paper, we chose to use feature scaling to normalise our data. Since the data consists of signed 16-bit integers, we found the minimum and maximum values to be -32,768 and 32,767 respectively and performed feature scaling to remap all data to be in the range [-1, 1].

We then used the provided data loader to separate the dataset into batches of 10 audio clips, each audio clip consisting of 10 separate sub-clips of approximate duration of 3 seconds.

B. Optimiser and Loss Function

Dieleman and Schrauwen omit some implementation details of the model in the original paper. Namely, the authors did not specify the optimiser and loss function used to train their model. For the optimiser, we chose Stochastic Gradient Descent (SGD) with momentum. The Subsection VI-C below describes how we selected the momentum value. Since our model is dealing with a classification problem, for our loss function we chose to use the Binary Cross Entropy (BCE) loss.

C. Hyperparameter Tuning

Once we settled on the final architecture, optimiser and loss function, we ran multiple experiments to help us tune the hyperparameters of the model. The two hyperparameters we tuned are the momentum and learning rate. Table I contains the results of our experiments. We consistently found that a high momentum value with a relatively low learning rate resulted in a higher AUC score. In particular, we found the combination of a momentum value of 0.99 and a learning rate of 0.005 resulted in the highest AUC score of 0.82. Section VII describes in detail the quantitative results of training our model with these parameters using different length and stride values.

VII. Replicating Quantitative Results

The results of our base implementation model can be seen in Table II. We ran our model on the same set of combinations of strides and lengths as in the original paper. Our AUC results follow a similar trend to that of Dieleman and Schrauwen [1]. We found that the AUC score increases for decreasing length and stride. In the original paper, the highest AUC was obtained for a length of 256 and a stride of 256. In our case, we obtained a marginally better result for the length of 512 and stride of 256.

After finalising the hyperparameter values, we evaluated our model against the test dataset with a length and stride of 256. Taking the average of the AUC score on the final epoch for five runs, our model achieved an average AUC score of 0.8267 with a standard deviation of 0.0027.

momentum	**learning rate**				
	0.001	0.002	0.005	0.01	0.02
0.99	0.76	0.79	0.82	0.67	0.50
0.97	0.73	0.75	0.79	0.65	0.51
0.95	0.69	0.73	0.77	0.67	0.50
0.90	0.68	0.70	0.74	0.72	0.66
0	0.51	0.51	0.63	0.72	0.50

TABLE I: Hyperparameter Tuning Results: AUC score on the Validation Dataset

VIII. Training Curves

Figure 3 and Figure 4 demonstrate the AUC and loss per epoch for our base model implementation respectively. As

length	**stride**	**AUC** (raw audio)
1024	1024	0.8052
1024	512	0.8192
512	512	0.8127
512	256	0.8282
256	256	0.8267

TABLE II: AUC Scores for Different Stride and Length on the Test Dataset

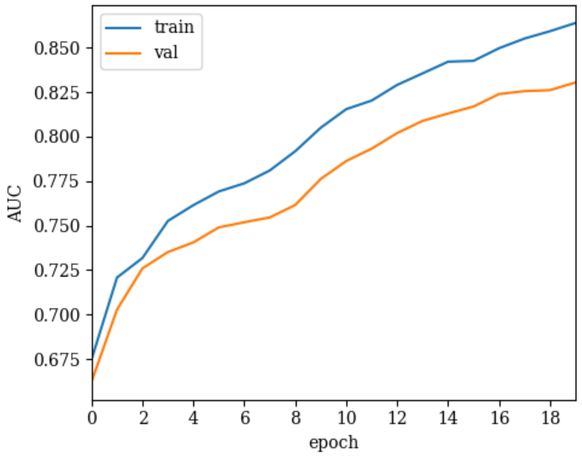

Fig. 3: AUC Curve for Training and Validation Data Sets

expected, during the first few epochs, the model scores a low AUC and a high average loss. During training, the AUC gradually increases and the loss decreases for both datasets, suggesting an improving predictive performance. Training for 20 epochs the model presents little to no overfitting. The divergence between the train and test curves increases suggesting that training the model for longer would result, to some extent, in overfitting and decreasing performance on the validation set.

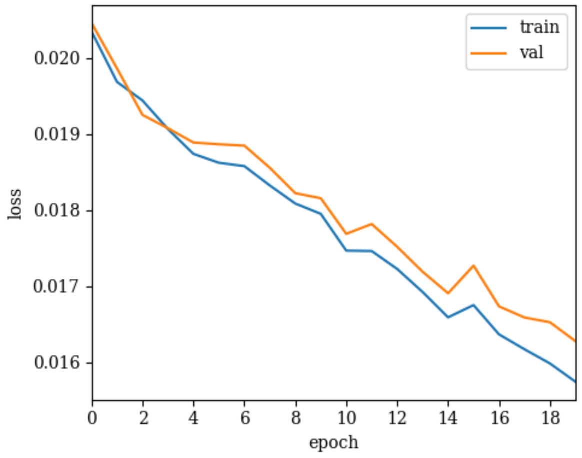

Fig. 4: Loss Curves for Training and Validation Data Sets

IX. Qualitative Results

A One-vs-the-Rest (OvR) method was applied to the testing dataset to calculate AUC scores for each tag. This allows us to gain a qualitative understanding of which genres the model does and does not perform well on. Figure 5 displays these results as well as the number of occurrences for each tag in the training dataset.

As shown by the plot, the model is best at classifying *heavy*, *rock*, and *hard*. One could consider these genres semantically similar, thereby implying that there may be common features shared between audio files annotated with these tags. This may further imply that the model is particularly good at classifying these common features. In fact, there are 133 clips tagged with all three of these tags meaning that 40.18% of *heavy* clips and 74.72% of *hard* clips are tagged with all three. A concrete example of an audio file from the training set which has all three of these tags and has a perfect AUC score of 1.0 is '*test/d/utopia_banished-night_of_the_black_wyvern-04-night_of_the_black_wyvern-349-378.npy*'.

With a large gap in AUC score compared to the next worst tags, the tags the model is worst at classifying are *no beat*, *no singer* and *no piano*. These are the only tags describing the absence of a feature: an unsurprising result. A possible explanation is that just the absence of one particular feature still allows that audio clip to contain features representative of other tags. For example, the audio file '*test/e/solace-balance-03-haiku-146-175.npy*' tagged as *no piano* achieved an extremely low AUC score of 0.4508.

Finally, there is a slight gender-based performance discrepancy. Music tagged as *male* has a slightly higher AUC score than *female* tagged music. A possibility is that this is a spurious correlation. As this is a multi-label dataset it could be that music tagged as male is more frequently associated with other types of music which happen to be easier to classify.

X. Improvements

A. Spectrograms

As shown by Dieleman and Schrauwen [1], despite the proposed CNN model performing highly on raw audio, it failed to outperform the model trained on spectrograms. Therefore, as part of our improvements to the model, we preprocessed the raw audio signal to convert each audio file to a mel-spectrogram. Mel-spectrograms are a time-frequency representation of audio signals, where the frequency is converted to the mel-scale, which has been shown to mimic the way humans perceive frequencies more accurately [7]. The parameters used to create the spectrograms follow the ones suggested by Panwar et al. [3]: Fast-Fourier-Transform (FFT) frame size of 512, hop size of 256, sampling rate of 12,000 and 96 mel filters. Finally, a zero-padding of 37 frames is added to both ends of the time axis.

B. Architecture and Learning

The architecture in our improved model consists of a CNN component and an RNN component. The CNN component consists of a 4-layer 2-dimensional CNN with a kernel size

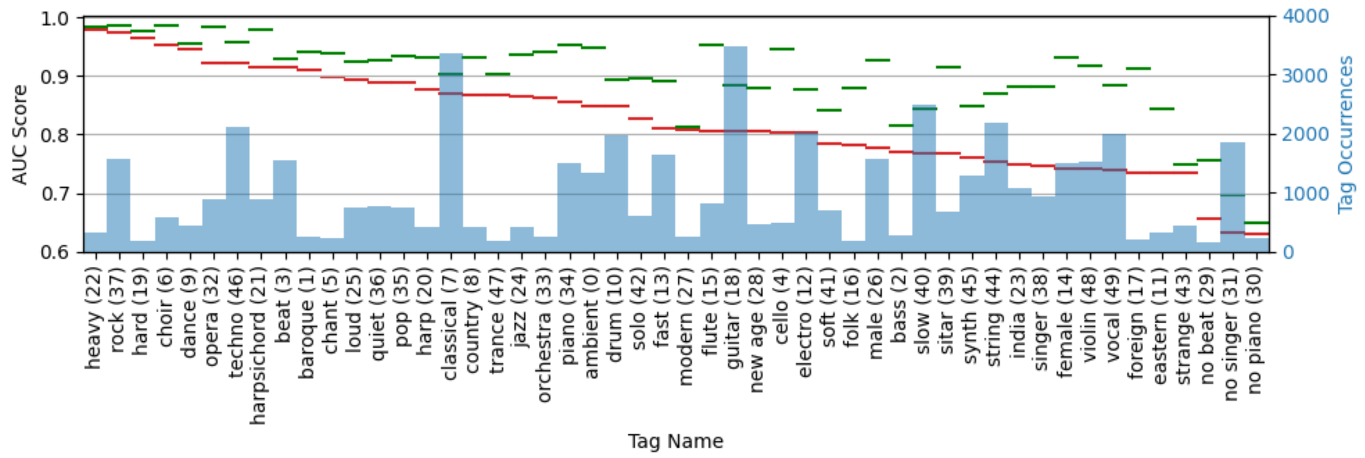

Fig. 5: AUC scores for each tag were calculated using a One-vs-the-Rest (OvR) method on the testing dataset. Red: AUC scores for CNN Architecture. Green: AUC scores for CRNN Architecture Blue: Corresponding number of occurrences for each tag within the training dataset. Plot format adapted from Choi et al. [2].

of 3×3 and a padding setting which maintains dimensionality. After each convolution, batch normalisation is applied, followed by the Exponential Linear Unit (ELU) activation and max pooling. The first two max pooling layers have a kernel size of 2×2 and 3×3 respectively, and the last two both have a kernel size of 4×4. The resulting output channels for each of the four convolution layers are then 64, 128, 128 and 128 respectively.

The signal is then flattened before being passed to the RNN component consisting of a 2-layer Gated Recurrent Unit (GRU), each one expecting and producing 15 input and output channels. After each layer, the ELU activation function is once again applied. Finally, the signal is passed to a single fully connected layer and the sigmoid function is applied to produce the output logits. Figure 6 shows a visual depiction of the architecture.

The optimiser used during training is Adam and the loss function is Binary Cross Entropy (BCE) loss as before. It is important to note that, while the model is still trained using batches, the raw audio clips (and therefore the spectrograms) are no longer split into sub-clips. This means that the model considers the whole duration of the audio file before making a prediction.

C. *Hyperparameter Tuning and Training Duration*

To achieve high performance in our improved model, it was essential to perform hyperparameter tuning. We chose five values of the learning rate and tested the improved model on the validation set. We found that the Adam optimiser prefers lower values of the learning rate. The results of our hyperparameter tuning can be seen in Figure 7. The AUC score increases rapidly during the first few epochs, after which, it begins to drop for some of the learning rate values. In particular, we found that when using a learning rate greater or equal to 0.0005, the AUC score on the validation set begins to deteriorate rapidly after approximately the 10th epoch. This is likely because the model begins to overfit. In contrast, we found that a very low learning rate of 0.0001 gives rise to a more steady curve which continues increasing throughout the 20 epochs of training. This suggests that, at this configuration, significant overfitting is yet to be seen. As a result, for our final improved model parameters, we chose to use a learning rate of 0.0001 and train the model for the full duration of 20 epochs.

D. *Quantitative Analysis*

As before, we tested our improved model against the test dataset, averaging the AUC score across five different runs. We were able to achieve a final average AUC score of 0.8949 with a standard deviation of 0.0028. This is an increase of 8.25% over our initial base model performance. This AUC score is also higher than that of Panwar et al. [3], whose model achieved 0.893, however, lower than that of Wang et al. [4], whose model was capable of reaching an AUC score of 0.9078. It is clear that using the two-dimensional mel-spectrograms has allowed the model to identify more complex features that helped increase its prediction accuracy. Furthermore, since the architecture of the improved model consists of more layers, the number of learnable weights is greater, which is also likely to be an explanation for the improvement in performance. Finally, it may also be the case that analysing the full duration of the audio track provides more temporal context for the model and enables it to make better predictions.

E. *Qualitative Analysis*

It is clear that the improved model performs better at the audio tagging task. Figure 5 shows that the improved model achieves higher AUC scores for every tag. In particular, we observed that the model performs significantly better at identifying *flute* and *female*. Due to this improvement, there is a reduced gap in gender-based tagging disparity. While there is no tag on which the new model performs worse, some tags only show marginal improvement. More specifically, the

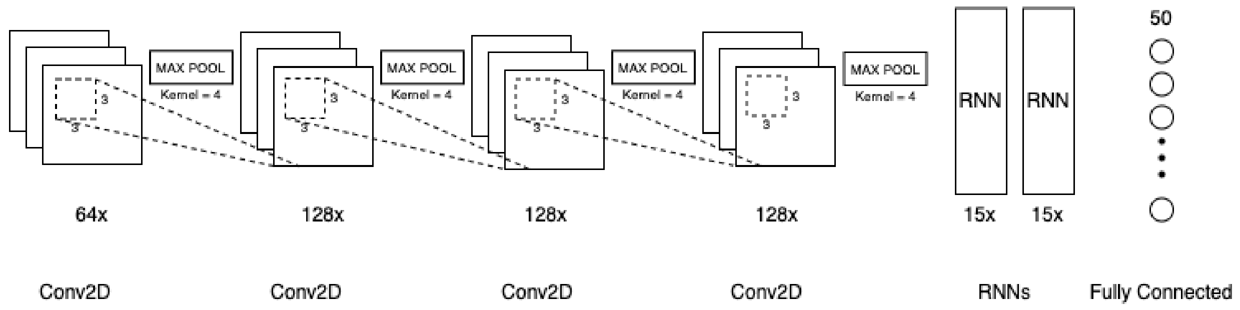

Fig. 6: The CRNN Architecture

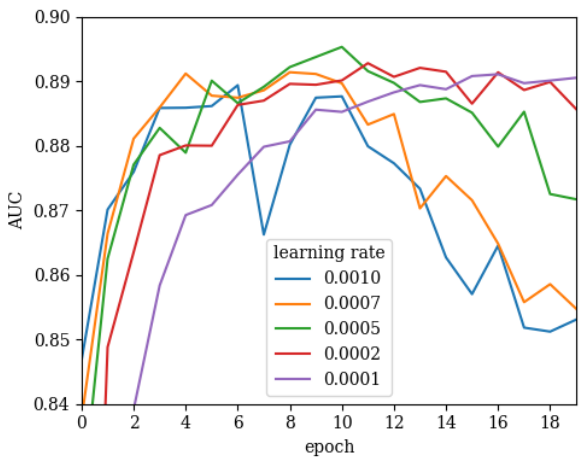

Fig. 7: AUC Curves for Validation Dataset on CRNN

heavy, *dance* and *modern*, show the least improvement out of all tags.

XI. Conclusion and Future Work

In this paper, we replicated the results from Dieleman and Schrauwen [1], implementing a CNN for an audio tagging task using the MagnaTagATune dataset. We were able to obtain an AUC-ROC score of 0.8267, outperforming the original model proposed. We empirically analysed the performance of the model and identified some of its weaknesses and strengths. We then presented an improved model based on the architecture proposed by Panwar et al. [3] trained on mel-spectrograms produced from the raw audio files. We were able to successfully show that a deeper, more complex model, analysing two-dimensional mel-spectrogram data was capable of achieving higher prediction performance. Our improved model achieved an AUC score of 0.8949, nominally higher than the authors' model.

Future work should be to incorporate additional known performance improvements into the model by implementing the architecture proposed by Wang et al. [4] discussed in further detail in Section III. Further expansions could include introducing a learning rate scheduler to adjust the learning rate during training and minimise any over-fitting. We would also suggest investigating in greater depth, how the limitations of the dataset discussed in Section IV, affect the performance of the model, and explain some of the phenomena discussed in the qualitative analysis in Section IX.

References

[1] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 6964–6968.

[2] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 2392–2396.

[3] S. Panwar, A. Das, M. Roopaei, and P. Rad, "A deep learning approach for mapping music genres," in *2017 12th System of Systems Engineering Conference (SoSE)*. IEEE, 2017, pp. 1–5.

[4] Z. Wang, S. Muknahallipatna, M. Fan, A. Okray, and C. Lan, "Music classification using an improved crnn with multi-directional spatial dependencies in both time and frequency dimensions," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[5] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: a platform for the creation of open audio datasets," in *Hu X, Cunningham SJ, Turnbull D, Duan Z, editors. Proceedings of the 18th ISMIR Conference; 2017 oct 23-27; Suzhou, China.[Canada]: International Society for Music Information Retrieval; 2017. p. 486-93.* International Society for Music Information Retrieval (ISMIR), 2017.

[6] G. Marques, F. Gouyon, T. Langlois, and M. A. Domingues, "Three current issues in music autotagging," in *2011 International Society for Music Information Retrieval Conference (ISMIR)*. International Society for Music Information Retrieval Conference (ISMIR), 2011.

[7] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The journal of the acoustical society of america*, vol. 8, no. 3, pp. 185–190, 1937.