MediaEval 2018 AcousticBrainz Genre Task: A CNN Baseline Relying on Mel-Features

Hendrik Schreiber
tagtraum industries incorporated, USA
hs@tagtraum.com

ABSTRACT

These working notes describe a relatively simple baseline for the MediaEval 2018 AcousticBrainz Genre Task. As classifier it uses a fully convolutional neural network (CNN) based on only the lowlevel AcousticBrainz melband features as input.

1 INTRODUCTION

We present a baseline approach for the MediaEval 2018 AcousticBrainz Genre Task. The task is defined as follows:

Based on provided track-level features, participants have to estimate genre labels for four different datasets (AllMusic, Discogs, Last.fm, and tagtraum), featuring four different label namespaces. Subtask 1 asks participants to train separately on each of the datasets and their respective labels and predict those labels for separate test sets. Subtask 2 allows training on the union of all four training datasets, but still requires predictions for the four test sets in their respective label spaces. For more details about the tasks see [1].

2 APPROACH

Our baseline approach explores how well a *convolutional neural network* (CNN) performs that has been trained on a relatively small subset of the available pre-computed features. For this purpose we have chosen to train only on Mel-features. The complete code is available on GitHub[1].

2.1 Feature Selection

Traditionally, *music genre recognition* (MGR) has often relied on Mel-based features—in fact, one of the most often cited MGR publications uses *Mel-frequency cepstral coefficients* (MFCCs) [10]. Mel-based approaches attempt to capture the timbre of a track, thus allowing conjectures about its instrumentation and genre. They do not necessarily take temporal properties into account and therefore often ignore an important aspect of musical expression, which can also be used for genre/style classification, see e.g., [8]. But since we are only interested in finding a baseline for more sophisticated systems, using just the provided `melbands` features is a reasonable approach. Lowlevel AcousticBrainz[2] data offers nine different Mel-features (global statistics: min, max, mean, ...) with 40 bands each, resulting in a total of 360 values per track. Because Mel-bands have a spatial relationship to each other, we organize the data into nine different channels, each featuring a 40-dimensional vector resulting in a $(N, 40, 9)$-dimensional tensor with N being the number of samples. Each of the 40-dimensional feature vectors is scaled so that its maximum is 1.

[1] https://github.com/hendriks73/melbaseline
[2] https://acousticbrainz.org/

MediaEval'18, 29-31 October 2018, Sophia Antipolis, France

Dataset	Number of Parameters
AllMusic	918,646
Discogs	685,479
Last.fm	691,683
tagtraum	675,656
Subtask 2	1,258,315

Table 1: Number of network parameters per dataset.

2.2 Neural Network

We choose to use the *fully convolutional network* (FCN) architecture depicted in Figure 1. In essence, the network consists of four similar feature extraction blocks, each formed by a one-dimensional convolutional layer, an ELU activation function [2], a dropout layer [9] with dropout probability 0.2, an average pooling layer (omitted in the last extraction block), and lastly a batch normalization layer [4]. From block to block the number of filters is increased from 64 to 512 as the length of the input decreases from 40 to 5 due to average pooling with a pool size of 2. The feature extraction blocks are followed by a classification block consisting of a one-dimensional convolution, an ELU activation function, a batch normalization layer, a global average pooling layer and sigmoid output units. The sigmoid activation function for the output is used, because the task is a multi-label multi-class problem. Note that the number of output dimensions depends on the number of different labels in the dataset. We therefore refer to it with the placeholder OUT. The total number of parameters in each networks is listed in Table 1.

2.3 Training

For subtask 1 we train the network using the provided training and validation sets with binary cross-entropy as loss function, Adam [5] with a learning rate of 0.001 as optimizer, and a batch size of 1,000. To avoid overfitting we employ early stopping with a patience of 50 epochs and use the last model that still showed an improvement in its validation loss.

Because the training data is very unbalanced, we experimented with balancing the training data with respect to the main genre labels via oversampling. As this led to worse results, balancing is not part of this submission.

For subtask 2 we gently normalize the provided labels by converting them to lowercase and removing all non-alphanumeric characters. Based on these transformed labels we create a unified training set.

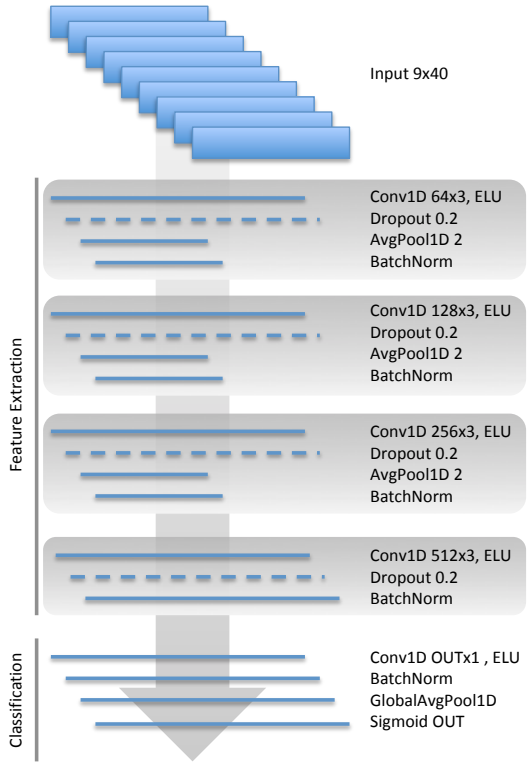

Figure 1: Schematic architecture of the neural network.

2.4 Prediction

The output of the last network layer consists of as many values in the range $[0, 1]$ as we have different labels in the dataset (OUT). If one of these values is greater than a predefined threshold, we assume that the associated label is applicable for the track. In order to optimize the tradeoff between precision and recall, we choose this threshold individually for each label based on the maximum F-score for predictions on the validation set [6], also known as *plug-in rule approach* [3]. In case the threshold is not crossed by any prediction for a given track, we divide all predictions by their thresholds and pick the label corresponding to the largest value.

Since we are using one unified training set for subtask 2, we need to reduce its output to labels that are valid in the context of a specific test dataset. We do so by reverting the applied normalization and dropping all labels not occurring in the test dataset.

3 RESULTS AND ANALYSIS

We evaluated a single run for both subtask 1 and 2. Results are listed in Tables 2 and 3. As expected, all results are well below last year's winning submission [6], which used a much larger network and 2,646 features. But the achieved scores are competitive with last year's second ranked submission [7], which used a similar number of features, though very different ones. Somewhat unexpected, the network trained for subtask 2 was not able to benefit from the additional training material and reaches generally slightly lower results than the networks trained on individual datasets for subtask 1.

Average per		Dataset			
		AllMusic	tagtraum	Last.fm	Discogs
Track (all labels)	P	0.292	0.3587	0.3707	0.3659
	R	0.4669	0.5074	0.4692	0.5436
	F	0.306	0.3918	0.374	0.3972
Track (genre labels)	P	0.6013	0.6149	0.5617	0.6937
	R	0.6777	0.6772	0.6318	0.7522
	F	0.6072	0.6271	0.5738	0.6902
Track (subgenre labels)	P	0.2031	0.256	0.228	0.2049
	R	0.3116	0.4135	0.3284	0.3662
	F	0.216	0.2922	0.2461	0.236
Label (all labels)	P	0.1141	0.1753	0.1993	0.1624
	R	0.1447	0.2213	0.2261	0.2115
	F	0.1148	0.1824	0.1977	0.1656
Label (genre labels)	P	0.3213	0.3444	0.3645	0.4523
	R	0.3384	0.3627	0.3812	0.4519
	F	0.3239	0.3467	0.3661	0.4466
Label (subgenre labels)	P	0.1083	0.1555	0.1826	0.1479
	R	0.1392	0.2047	0.2105	0.1995
	F	0.1089	0.1632	0.1807	0.1515

Table 2: Precision, recall and F-scores for subtask 1.

Average per		Dataset			
		AllMusic	tagtraum	Last.fm	Discogs
Track (all labels)	P	0.285	0.359	0.3411	0.3563
	R	0.4713	0.5079	0.4773	0.544
	F	0.3041	0.385	0.354	0.3877
Track (genre labels)	P	0.5923	0.594	0.5068	0.6801
	R	0.6762	0.678	0.6434	0.7484
	F	0.6011	0.6127	0.5413	0.6802
Track (subgenre labels)	P	0.2071	0.2486	0.2021	0.1959
	R	0.3198	0.4131	0.3282	0.37
	F	0.2205	0.2825	0.2261	0.229
Label (all labels)	P	0.1127	0.1662	0.1703	0.1526
	R	0.1466	0.2272	0.2176	0.2138
	F	0.1141	0.1797	0.1763	0.1619
Label (genre labels)	P	0.3174	0.3291	0.3304	0.4491
	R	0.3395	0.363	0.3874	0.4454
	F	0.3191	0.3398	0.3484	0.4407
Label (subgenre labels)	P	0.1069	0.1471	0.1541	0.1378
	R	0.1412	0.2114	0.2005	0.2022
	F	0.1083	0.161	0.1589	0.1479

Table 3: Precision, recall and F-scores for subtask 2.

4 DISCUSSION AND OUTLOOK

We have shown that using a relatively small and simple convolutional neural network (CNN) trained only on global Mel-features can achieve respectable scores in this task. Adding temporal features may improve the results further.

REFERENCES

[1] Dmitry Bogdanov, Alastair Porter, Julián Urbano, and Hendrik Schreiber. 2018. The MediaEval 2018 AcousticBrainz Genre Task: Content-based Music Genre Recognition from Multiple Sources. In *MediaEval 2018 Workshop*. Sophia Antipolis, France.

[2] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus), In International Conference on Learning Representations (ICLR). *arXiv preprint arXiv:1511.07289*.

[3] Krzysztof Dembczynski, Arkadiusz Jachnik, Wojciech Kotlowski, Willem Waegeman, and Eyke Hüllermeier. 2013. Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *International Conference on Machine Learning*. Atlanta, GA, USA, 1130–1138.

[4] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint arXiv:1502.03167* (2015).

[5] Diederik P. Kingma and Jimmy Lei Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[6] Khaled Koutini, Alina Imenina, Matthias Dorfer, Alexander Gruber, and Markus Schedl. 2017. MediaEval 2017 AcousticBrainz Genre Task: Multilayer Perceptron Approach. In *MediaEval 2017 Workshop*. Dublin, Ireland.

[7] Benjamin Murauer, Maximilian Mayerl, Michael Tschuggnall, Eva Zangerle, Martin Pichl, and Günther Specht. 2017. Hierarchical Multi-label Classification and Voting for Genre Classification. In *MediaEval 2017 Workshop*. Dublin, Ireland.

[8] Björn Schuller, Florian Eyben, and Gerhard Rigoll. 2008. Tango or Waltz?: Putting Ballroom Dance Style into Tempo Detection. *EURASIP Journal on Audio, Speech, and Music Processing* 2008 (2008), 12.

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[10] George Tzanetakis and Perry Cook. 2002. Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing* 10, 5 (2002), 293–302.