Single and Multi Column Neural Networks for Content-based Music Genre Recognition

Chang Wook Kim[2], Jaehun Kim[3], Kwangsub Kim[1], Minz Won[1*]
[1]Kakao Corp., Republic of Korea
[2]Kakao Brain, Republic of Korea
[3]Delft University of Technology, Netherlands

ABSTRACT

This working note reports approaches of team KART to MediaEval2017 AcousticBrainz Genre Task and their results. To solve the problem, we mainly considered the sparsity and noise of data, network design for the multi-label classification, and implementation of successful Deep Neural Network (DNN) models. We propose three steps of preprocessing and depict two different approaches: a single-column model and a multi-column model.

1 INTRODUCTION

A music genre is a class, type or category that defined by convention [6]. However, taxonomies of music genres can differ by communities. The MediaEval2017 AcousticBrainz Genre Task aims to predict the genre and subgenre of unlabeled music recordings from four different datasets which consist of four different genre/subgenre taxonomies [1]. Each dataset includes precomputed music audio features using Essentia library [2] and genre/subgenre annotations that follow its own taxonomy.

We approached the problem based on careful consideration of following:

- How to handle the noisy and sparse data?
- How to solve the multi-label classification task?
- How to apply a variety of successful deep neural network models to our task?

2 PREPROCESSING

Before starting the model training, we conducted three steps of feature preprocessing: (*i*) feature vectorization, (*ii*) fitting outlier feature values to the outlier boundaries, and (*iii*) selecting features by feature value distribution analysis.

Essentially, we tried to use features as raw as possible. Its underlying assumption is that the deep neural network model can learn useful representations of the raw data if there are sufficient amount of samples. We omitted all the information under 'meta data' keys. Further, we applied PCA to the covariance matrices of filter banks. For the computation convenience, we only choose the eigenvector whose corresponding eigenvalue is most large. In addition, we encoded categorical features into binary vectors in one-hot manner.

For some features, there are outliers with extremely high or low values in comparison with their medians (Figure 1 left). We suppressed or boosted these outlier values to the outlier boundaries. A lower limit and an upper limit boundaries for judging outliers are defined as :

$$\begin{aligned} LowerLimit &= FirstQuartile - 1.5 * IQR \\ UpperLimit &= ThirdQuartile + 1.5 * IQR \end{aligned} \tag{1}$$

where Inter Quartile Range (IQR) is a difference between the first quartile and the third quartile. Values bigger than the upper limit were clipped at the upper limit, and values smaller than the lower limit were boosted to the lower limit. Figure 1 (right) shows the distribution after the fitting of the Figure 1 (left).

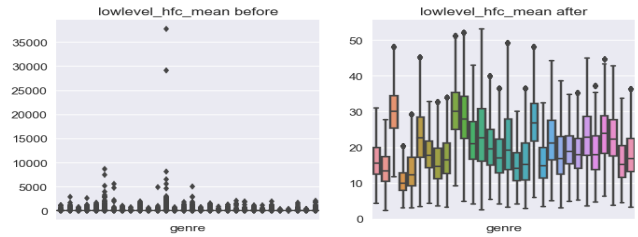


Figure 1: Box plots of High Frequency Content (HFC) mean for 30 genres before clipping (left) and after clipping (right).

Upper limits and lower limits of the training set were derived from feature value distributions of each feature and each genre. However, since the genre labels of the test set should not be informed, we thresholded the test set at the maximum upper limit and the minimum lower limit of the training set.

After the outlier fitting, we defined features that concentrate around same values for every genre as useless features and removed 133 features from the training and test sets.

3 MODEL

We implemented two Feed-forward Neural Networks (FNNs). The main difference in architectures is whether the label hierarchy between the genre and the sub-genre is considered explicitly. Since the provided input features are already processed, the model is designed for encoding interdependency among labels.

3.1 Single Column Model

As a baseline, we implemented an Single Column FNN (SCNN) whose output dimensions correspond to the entire labels. The label hierarchy between genre and sub-genre was not considered in this model. The genre and sub-genre are equally treated as independent labels. We applied the weight vector $\mathbf{w}$ with the loss function, to

*Author's names are listed alphabetically; authors contributed equally to this work.

MediaEval'17, 13-15 September 2017, Dublin, Ireland

give less penalty for more frequent labels as following:

$$\mathbf{w}_i = 1 + \frac{1}{\log(1 + \mathbf{f}_i)} \tag{2}$$

where $\mathbf{f}_i$ denote raw count of label i in the given training dataset. In this way, the error from less frequent labels can be counted relatively larger than more frequent labels. It leads to learning less frequent labels more sensitively. The loss function of this model is:

$$L_{SCNN} = \frac{1}{M} \sum_{m,i} \mathbf{w}_i H(\hat{\mathbf{y}}_i^{(m)}, \mathbf{y}_i^{(m)}) \tag{3}$$

where H denotes the *binary cross-entropy* of the true label and the prediction, $\mathbf{y}_i^{(m)}$ is a binary vector for label i of the observation m, $\hat{\mathbf{y}}_i^{(m)}$ is a prediction for label i of the observation m, and M denotes the size of the mini-batch. The inference of $\hat{\mathbf{y}}_i^{(m)}$ is:

$$\hat{\mathbf{y}}^{(m)} = f(\mathbf{x}^{(m)}; \theta) \tag{4}$$

where $f(\mathbf{x}^{(m)}; \theta)$ denotes an FNN that has a set of model parameters denoted as θ, and $\mathbf{x}^{(m)}$ is a feature vector of the observation m. We applied ReLU[5] activation function for hidden layers and used the sigmoid function for the output layer. We also used the dropout[7] for every hidden layers with the dropout probability 0.5. We applied the batch normalization[4] to the hidden layers to accelerate training process. The details are depicted in Figure 2.

During the inference, we only predict labels whose probability exceed the threshold α. We set α as 0.2 and 0.3 for SCNN, which are found as optimal values by the cross-validation.

3.2 Multi Column Model

To explicitly reflect label dependency between sub-genre and genre, we implemented a Multi Column FNN model (MCNN). It has a parallel structure for each set of sub-genre and genre, and merged by the Bayes rule on the top layer, as following:

$$\hat{\mathbf{y}}_g^{(m)} = f(\mathbf{x}^{(m)}; \theta_g) \tag{5}$$

$$\hat{\mathbf{y}}_{sg}^{(m)} = \hat{\mathbf{y}}_{g^*}^{(m)} f(\mathbf{x}^{(m)}; \theta_{sg}) \tag{6}$$

where $\hat{\mathbf{y}}_g^{(m)}$ and $\hat{\mathbf{y}}_{sg}^{(m)}$ denote the estimated probabilities of genre and sub-genre from each model. The posterior probability of sub-genre $\hat{\mathbf{y}}_{sg}^{(m)}$ is conditioned by $\hat{\mathbf{y}}_{g^*}^{(m)}$. Here g^* denotes the genre where the sub-genre sg belongs. The loss function of this model is:

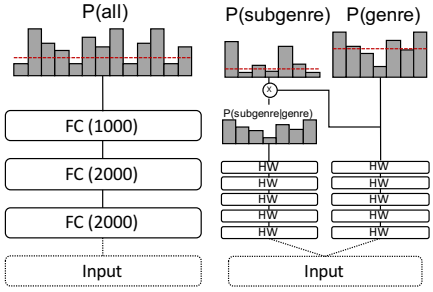
$$L_{MCNN} = \frac{1}{M} \sum_{m} [w_g H(\hat{\mathbf{y}}_g^{(m)}, \mathbf{y}_g^{(m)}) + w_{sg} H(\hat{\mathbf{y}}_{sg}^{(m)}, \mathbf{y}_{sg}^{(m)})] \tag{7}$$

where w_g and w_{sg} are scalar weights to balance learning rate of the *genre column* and the *sub-genre column*. We used the ratio of 1:9 between w_g and w_{sg}, considering sub-genre labels are more sparse than genre labels. We used the batch normalization only after the input layer. The dropout was not applied. We used threshold $\alpha_{sg} = 0.25$ for sub-genre and threshold $\alpha_g = 0.4$ for genre, respectively.

Assuming given feature set is already sufficiently processed, we also applied a highway network[8] architecture. It controls the gradient flow by the parametric gate at each layer, similar to the Long Short-Term Memory[3]. We applied this architecture for the network not only to have a deeper structure, but also to use the information more close to the input feature.

Table 1: Mean F1 scores on test set: F1 values averaged over all datasets

Runs	$F1_{track}$	$F1_{label}$
Baseline1	0.1095	0.007
Baseline2	0.2378	0.003
SCNN	**0.2526**	**0.0085**
MCNN	0.1828	0.0084



(a) Single-Column Model (b) Multi-Column Model

Figure 2: The architectures of suggested models. (left) The SCNN, which has 3 hidden fully-connected (FC) layers and one output layer. The number inside the parentheses indicates the number of units in each layer. (right) The structure of the MCNN, which has 5 highway blocks (HW) and an output layer. The number of units in each highway blocks are identical to the input dimensionality. The red dotted lines indicate the threshold for each model.

4 RESULTS AND ANALYSIS

A partial result of our runs and baselines obtained from test set presented in Table 1. The scores are mean $F1$ scores *per tracks* and *per labels*, respectively, which are averaged over all results from datasets. Baseline1 is a random predictor and Baseline2 is a majority predictor. SCNN presented in Table 1 uses threshold $\alpha = 0.2$.

When comparing the scores, we noticed that SCNN is overall better than Baselines, but the MCNN is better than Baselines only in *per label* scores. However, considering the recall scores, which are not presented in the note due to space, it shows that both suggested models score better recall than the Baseline 2. This shows both models are working better for predicting sparse sub-genres than baselines and suggesting our weighted losses work as intended.

Compared to the validation accuracy, the test accuracy got worse. Since the training set and the validation set are skewed and sparse data, our models failed to learn generalized parameters. Experiments with the data augmentation have to be explored to overcome the drawback.

Also, a large model size of MCNN can be another reason of its worse test accuracy. The highway networks have 2 times larger than the standard fully-connected layers and MCNN has two columns of the network to model genre and sub-genre predictor separately. This structure makes the model 4 times bigger than SCNN model. A Multi-Column architecture with small units and standard fully-connected layers will be useful.

REFERENCES

[1] Dmitry Bogdanov, Alastair Porter, Julian Urbano, and Hendrik Schreiber. 2017. MediaEval 2017 AcousticBrainz Genre Task: Content-based Music Genre Recognition from Multiple Sources.

[2] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, José R Zapata, Xavier Serra, and others. 2013. Essentia: An Audio Analysis Library for Music Information Retrieval. In *International Society for Music Information Retrieval (ISMIR'13) Conference*. Curitiba, Brazil, 493–498.

[3] Sepp Hochreiter and JÃijrgen Schmidhuber. 1997. Long Short-term Memory. 9 (12 1997), 1735–80.

[4] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*. 448–456.

[5] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

[6] Jim Samson. 2017. Genre. In *Grove Music Online. Oxford Music Online*. Oxford University Press. Web., http://www.oxfordmusiconline.com/subscriber/article/grove/music/40599.

[7] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15, 1 (2014), 1929–1958.

[8] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387* (2015).