Wide Ensembles of Neural Networks in Music Genre Classification

Daniel Kostrzewa, Wojciech Mazur, and Robert Brzeski(✉)

Department of Applied Informatics, Silesian University of Technology, Gliwice, Poland
{daniel.kostrzewa,robert.brzeski}@polsl.pl

Abstract. The classification of music genres is essential due to millions of songs in online databases. It would be nearly impossible or very costly to do this job manually. That is why there is a need to create robust and efficient methods that automatically help to do this task. In this paper, music genre recognition is implemented by exploiting the potential of wide ensembles of neural network classifiers. Creating infrequently used types of ensembles is a main contribution of authors in the development of automatic recognition of the musical genre. The paper shows how it can be done in a relatively quick and straightforward manner. The presented method can be implemented in many other use cases.

Keywords: Classification · Machine learning · Music genre recognition · Wide ensemble · Free music archive dataset · Neural network

1 Introduction

There are millions of songs available for users in online databases. Very often, we would like to listen only songs that belong to a specified music genre. It is nearly impossible to manually assign these millions songs into a music genre. One of the options is to do this automatically. For this task, machine learning methods can be used. It is possible to improve the obtained classification quality, either by pre-processing the dataset or by appropriate selection of classifiers parameters, or by creating an appropriate classifier structure (e.g., the number of layers). However, for a given classifier, at some point, the practical ability for further improving the results is limited.

In the current research, we would like to examine the comparatively simple method – the collection of ensembles. Of course, it is possible to create an ensemble with quite complex classifiers, including a convolutional neural network with many layers [6]. However, the computational cost of using such an ensemble can

This work was supported by research funds for young researchers of the Department of Applied Informatics, Silesian University of Technology, Gliwice, Poland (grant no. 02/0100/BKM22/0021 – DK) and Statutory Research funds of the Department of Applied Informatics, Silesian University of Technology, Gliwice, Poland (grant no. 02/0100/BK22/0017 – RB).

D. Groen et al. (Eds.): ICCS 2022, LNCS 13351, pp. 64–71, 2022.
https://doi.org/10.1007/978-3-031-08754-7_9

be high. Deep neural networks take a while to learn. That is why, in current research, we intend to examine the creating of wide ensembles with relatively simple classifiers. Wide ensembles are understood as built with many (dozens) base-level classifiers, in this case, multiple copies of the same classifier. This way, we can obtain another method of improving the final result of the music genres classification.

The main contribution of the paper is the way of creating wide ensembles. It can be done instantly for a neural network by creating multiple copies of the previously prepared classifier. This paper will check the change in the classification quality for this type of structure. The second contribution is checking whether the additional, late input of raw data, connected directly to the concatenation layer, which connects individual classifiers, improves classification quality. Additionally, the influence of depth of the classifiers and application of Principal Component Analysis on the final result is examined.

2 Related Work

The problem of music genre recognition (MGR) [1], as one of the sub-disciplines of music information retrieval, has become an increasingly explored issue in recent years. The article [12] can be considered the beginning of the popularity of the MGR topic [5]. The classification of musical songs can be executed using many machine learning methods. Not only the classical classifiers [2] can be used, but also newer approaches like the deep learning domain [4], with convolutional neural networks (CNN) [8] or convolutional recurrent neural networks (CRNN) [11]. Unfortunately, deep neural networks are more challenging to create, need more time for the learning process, and often give worse classification results (comparative studies are presented in Table 4) or at least there is no guarantee for obtaining a better. There are also studies in which the ensembles of various classifiers are used. Ensembles consist of base-level with a set of classifiers, as well as meta classifier [10] that tries to predict the final result based on outcomes of base-level classifiers.

3 Conditions of Experiments

The dataset that the experiment was conducted is the small subset of Free Music Archive dataset (FMA) [3]. For each excerpt, there are over 500 features in the FMA dataset. This dataset was split into three sets: training, validation, and testing in ratio 80:10:10.

The ensemble in the conducted research is built as multiple copies of the same classifier. However, each of the classifiers is learned independently. Because they have various sets of initial weights, the result of the learning process is also different for each of the classifiers. Consequently, they will generate slightly different classification outcomes. The output of individual classifiers of a given ensemble is fed to the input of the 'Concatenate' layer of the additional classifier (based on a dense neural network), which generates the final classification result. Additionally, this layer can have neurons for extra input of numerical data.

4 Experiments

4.1 Basic Dense Classifiers

At the beginning, three basic classifiers were created. Their quality (Table 1) can be treated as a benchmark for further tests.

The first one is a **simple dense classifier** (Fig. 1a) consisting of three layers – input layer and two dense layers. The results can be found in Table 1 in the 'En0' row. Accuracy of 53% is far better than a blind guess and already looks promising. Each created ensemble has a unique name (number) from En1 up to En14. The current classifier (En0) is the only one which is not an ensemble.

Table 1. Different size of ensemble, without and with raw data input.

Name	Classifiers	Raw data input	Accuracy	Precision	Recall	F1-score
En0	1 (simplest classifier)	No	0.530	0.524	0.530	0.526
En1	2	No	0.539	0.540	0.542	0.540
En2	2	Yes	0.560	0.553	0.557	0.555
En3	3	Yes	0.541	0.538	0.537	0.535
En4	5	Yes	0.586	0.584	0.590	0.585
En5	50	Yes	**0.621**	**0.618**	**0.621**	**0.617**

The second structure – En1, is the **simplest ensemble that consists of only two classifiers, without numerical input** connected by predicating layer with classifier outputs merging mechanism. Those two classifiers are just a copy of the simplest classifier.

Next structure En2, **two classifiers ensemble with numerical input** (Fig. 1b) is almost the same as the previous one. However, the difference is that a raw numerical input is attached to the merging layer.

Comparing the obtained results (Table 1), it turns out that the best approach is the usage of the two classifiers ensemble with numerical input. An additional late raw numerical input (concatenated with outcomes of base-level classifiers) to the two classifiers network ensemble, significantly increased the classification quality of a whole structure. As the results came out to be quite promising, the additional layer of numerical input will be included for the rest of the research.

4.2 Principal Component Analysis Influence

In this part of the research, the Principal Component Analysis (PCA) was introduced. The feature vector can contain not only useful data but also noise and that can lead to worsening results. One of the popular methods of preventing such a problem is feature extraction. Over five hundred features of raw data are currently fed to classifiers. Reducing that number might not only speed up processing time but also increase the accuracy of the proposed network. Time

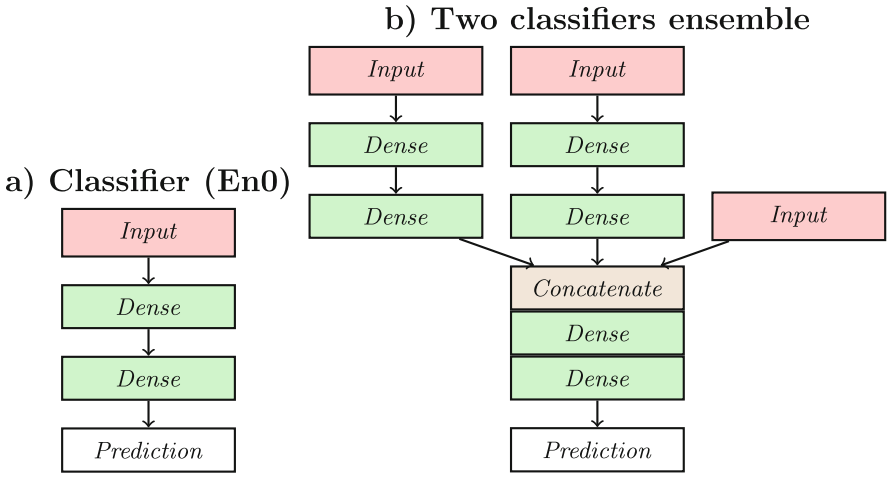

Fig. 1. a) Simplest classifier b) Ensemble with numerical input of raw data

for training for all input data without using PCA took around 6.5 s, while with PCA reduction to 300 components, training took 4 s, and with 20 components, training took only 2.8 s. The best result was achieved (Table 2) for 300 components, and that quantity will be used for the rest of the research. The reduced by PCA set of features are transferred to the classifiers' inputs and for the late raw numerical input.

Table 2. Accuracy achieved for Principal Component Analysis.

	Number of components						
Without PCA	500	400	300	200	100	50	20
0.553	0.555	0.549	**0.560**	0.553	0.548	0.521	0.486

4.3 Influence of the Number of Classifiers

In this part of the research, the influence of the number of classifiers was tested. The three (En3), five (En4), and fifty (En5) classifiers in one ensemble were taken into consideration. The achieved results are presented in Table 1.

It turns out that one additional classifier in the ensemble (En3) did not bring higher results. For another two classifiers added to the model (En4), the results are slightly improved. However, the best result is achieved for a wide ensemble of 50 base-level classifiers (En5), with an accuracy of about 62%. Nevertheless, better results also came with around 20 times longer training time than in the case of two base-level classifiers network. Additionally, the overfitting of the network can be seen. As a result, in the next test, batch normalization and dropout will be introduced.

Table 3. Different size of ensemble and different size of classifier.

Name	Classifiers	Architecture	Accuracy	Precision	Recall	F1-score
En6	2	3 Dense, batch norm.	0.578	0.576	0.570	0.572
En7	2	4 Dense, batch norm.	0.573	0.562	0.568	0.563
En8	5	3 Dense, batch norm.	0.551	0.545	0.553	0.547
En9	5	4 Dense, batch norm.	0.596	0.597	0.598	0.595
En10	50	3 Dense, batch norm.	**0.658**	**0.656**	**0.653**	**0.653**
En11	50	3 Dense, batch norm., dropout	0.601	0.603	0.607	0.598
En12	100	3 Dense, batch norm., dropout	0.609	0.610	0.613	0.610
En13	200	3 Dense, batch norm.	0.591	0.588	0.586	0.583
En14	200	3 Dense, batch norm., dropout	0.592	0.590	0.588	0.584

4.4 Influence of the Architecture of Base-Level Classifiers

This time not only the number but also the size and structure of classifiers were examined. The results of using such ensembles are presented in Table 3.

The first structure (En6) goes back to the two base-level classifiers with an additional dense and batch normalization. Comparing the obtained outcomes to the ensemble without additional layer and batch normalization (En2) shows a slight improvement in performance quality. In the next ensemble (En7), another dense layer and batch normalization were added. Apart from longer training time, there was no significant change in the classification accuracy by introducing the next dense layer. The structure of the En8 ensemble is similar to the En6, but this time with five base-level classifiers. Interestingly, the quantitative outcomes are slightly worse compared to both ensemble En6 and the earlier En4. However, the En9 ensemble, with another dense layer and batch normalization, easily beat all presented ensembles but En5 (with 50 base-level classifiers). The En10 wide ensemble is similar to the En9 one, however, this time consists of fifty classifiers. The obtained results are the best in the presented studies. A test was also carried out using additional dropout, but the results obtained in this way (En11, En12, and En14), turned out to be worse than En10. The same conclusion is for a wider ensemble (En13) with 200 classifiers but without dropout.

5 Comparison of the Outcomes

Basic classifiers and raw data input. As can be seen in Table 1 even the basic ensemble (En1) improves the results slightly in comparison to the simple classifier (En0). A much more significant improvement is obtained with a late raw numerical data input (En2).

Principal Component Analysis. Introduction of Principal Component Analysis (Table 2) was not strictly related to the model development but to the data preprocessing that the model operated on. The FMA dataset offers an overall of

518 features (for each music track), and data can be preprocessed by the dimensionality reduction method. Here, only PCA was exploited and the best result was achieved for 300 features.

Width of ensemble. Additional base-level classifiers have influenced a significant increase in quantitative results (Table 1). The actual cost of such an increase was only the time of computation. An increase of ensemble width by adding classifiers improved results significantly (up to fifty base-level classifiers). Further expanding the ensemble did not bring any advance, conversely, the outcomes have already started to worsen.

Architecture of base-level classifier. The next way of improving the accuracy of the model was the change in the structure of the base-level classifier (Table 3). Incrementing the depth of the base-level classifier results in higher accuracy values. At the same time, with more layers, the overfitting of the model became more noticeable. To reduce training accuracy spiking, batch normalization and dropout were introduced. Nonetheless, dropout did decrease overfitting but did not help with model accuracy.

Comparison of the quantitative outcomes with state-of-the-art. To compare the best result achieved in this research (wide ensemble En10 with fifty base-level classifiers for which accuracy was 0.658) with other state-of-the-art works [7,9,11,13–15] the Table 4 was created.

Table 4. Comparison of different models classifying FMA small dataset with the proposed wide ensemble En10 (all values are in %).

No.	Model	Accuracy	No.	Model	Accuracy
1	K-Nearest Neighbors [15]	36.4	12	MoER [14]	55.9
2	Logistic Regression [15]	42.3	13	FCN [13]	63.9
3	Multilayer Perceptron [15]	44.9	14	TimbreCNN [13]	61.7
4	Support Vector Machine [15]	46.4	15	End-to-end [13]	61.4
5	Original spectrogram [14]	49.4	16	CRNN [13]	63.4
6	Harmonic spectrogram [14]	43.4	17	CRNN-TF [13]	64.7
7	Percussive spectrogram [14]	50.9	18	CRNN [11]	53.5
8	Modulation spectrogram [14]	55.6	19	CNN-RNN [11]	56.4
9	MFCC [14]	47.1	20	CNN TL [7]	51.5
10	MoEB [14]	54.1	21	CNN TL [9]	56.8
11	MoEC [14]	55.6	22	C-RNN [15]	65.2
23	**Wide ensemble En10**	**65.8**	**23**	**Wide ensemble En10**	**65.8**

6 Conclusions

The results of the work are auspicious. The ensemble provided satisfying results, with the best model reaching almost 66% accuracy. It is worth mentioning that

this value is in the range of state-of-the-art techniques. However, obtaining such a result is only an additional effect. The main aim of the research was to show how to relatively easily improve the originally obtained classification result. This goal has been achieved. This way, by implementing wide ensembles, we obtain another method of improving the final result of the classification without much design or programming effort. A certain limitation may be the increased computation time and the increased demand for computer resources. However, there are no initial restrictions as to the field, dataset, or nature of the research where we can try to use this method.

Summarising, if the main goal is classification quality, presented in the article methods and structures are definitely worth considering.

References

1. Aucouturier, J.J., Pachet, F.: Representing musical genre: a state of the art. J. New Music Res. **32**(1), 83–93 (2003)
2. Basili, R., Serafini, A., Stellato, A.: Z classification of musical genre: a machine learning approach. In: ISMIR (2004)
3. Defferrard, M., Benzi, K., Vandergheynst, P., Bresson, X.: FMA: a dataset for music analysis. arXiv preprint arXiv:1612.01840 (2016)
4. Kereliuk, C., Sturm, B.L., Larsen, J.: Z deep learning and music adversaries. IEEE Trans. Multim. **17**(11), 2059–2071 (2015)
5. Knees, P., Schedl, M.: A survey of music similarity and recommendation from music context data. ACM Trans. Multim. Comput. Commun. Appl. **10**(1), 1–21 (2013)
6. Kostrzewa, D., Kaminski, P., Brzeski, R.: Music genre classification: looking for the perfect network. In: Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A. (eds.) ICCS 2021. LNCS, vol. 12742, pp. 55–67. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77961-0_6
7. Lee, D., Lee, J., Park, J., Lee, K.: Z enhancing music features by knowledge transfer from user-item log data. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 386–390. IEEE (2019)
8. Lim, M., et al.: Z convolutional neural network based audio event classification. KSII Trans. Internet Inf. Syst. **12**(6) (2018)
9. Park, J., Lee, J., Park, J., Ha, J.W., Nam, J.: Z representation learning of music using artist labels. arXiv preprint arXiv:1710.06648 (2017)
10. Silla, Jr., C.N., Kaestner, C.A.A., Koerich, A.L.: Automatic music genre classification using ensemble of classifiers. In: 2007 IEEE International Conference on Systems, Man and Cybernetics, pp. 1687–1692 (2007)
11. Snigdha, C., Kavitha, A.S., Shwetha, A.N., Shreya, H., Vidyullatha, K.S.: Z music genre classification using machine learning algorithms: a comparison. Int. Res. J. Eng. Technol. **6**(5), 851–858 (2019)
12. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Trans. Speech Audio Process. **10**(5), 293–302 (2002)
13. Wang, Z., Muknahallipatna, S., Fan, M., Okray, A., Lan, C.: Z 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), pp. 1–8. IEEE (2019)

14. Yi, Y., Chen, K.Y., Gu, H.Y.: Z mixture of CNN experts from multiple acoustic feature domain for music genre classification. In: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1250–1255. IEEE (2019)
15. Zhang, C., Zhang, Y., Chen, C.: Z songnet: Real-time music classification. Stanford University Press (2019)