Music Classification using Extreme Learning Machines

Simone Scardapane*, Danilo Comminiello*, Michele Scarpiniti* and Aurelio Uncini*
*Department of Information Engineering, Electronics and Telecommunications (DIET),
"Sapienza" University of Rome,
Via Eudossiana 18, 00184, Rome.
Emails: {simone.scardapane, danilo.comminiello, michele.scarpiniti}@uniroma1.it; aurel@ieee.org

***Abstract*—Over the last years, automatic music classification has become a standard benchmark problem in the machine learning community. This is partly due to its inherent difficulty, and also to the impact that a fully automated classification system can have in a commercial application. In this paper we test the efficiency of a relatively new learning tool, Extreme Learning Machines (ELM), for several classification tasks on publicly available song datasets. ELM is gaining increasing attention, due to its versatility and speed in adapting its internal parameters. Since both of these attributes are fundamental in music classification, ELM provides a good alternative to standard learning models. Our results support this claim, showing a sustained gain of ELM over a feedforward neural network architecture. In particular, ELM provides a great decrease in computational training time, and has always higher or comparable results in terms of efficiency.**

I. Introduction

The increased availability of musical content and user-generated annotations associated to that content has made Automatic Music Retrieval (AMR) a tool of fundamental importance for music applications. As an example Spotify[1], one of the biggest web applications for music streaming, announced last year to have reached an overall catalog of more than 20 million songs. Selecting songs from this database to provide a good experience to the end users results extremely challenging. AMR is hence the problem of efficiently retrieving songs that may be of interest to the end users depending on a given set of predefined criteria.

Automatic Music Classification (AMC) is one of the main problems in AMR. Clearly, as long as we are able to correctly classify a set of songs, we can use the resulting groups as a tool to satisfy a user-defined query. Each song may be classified according to several dimensions of interest, including genre, perceived mood, artist, presence of a given instrument, and several others. Fu et al. [1] provides an interesting overview of the field, by reviewing most of the relevant papers and techniques. Despite all the efforts, however, results are still far from being optimal, due to the inherent difficulty of the problem. Consider for example the following aspects:

1) A standard audio file comprises several thousands of samples, subdivided in one, two or more channels. Despite there exists a large set of possible features that can be extracted from a single track, it is a difficult task to select an optimal subset with respect to the task at hand. We delve into this point in more detail in Section II.
2) In some cases, the task may be challenging also for a human expert, due to the high degree of subjectivity and required knowledge involved. This is evident, for example, in the case of efficient genre classification.
3) Moreover, good accuracy may require large databases of thousands of songs. This results in several gigabytes of data to be elaborated, hence imposing a strong computational effort for the training of the learning models.

All these aspects are worsened when we include in our data user-generated content relative to each song. Consider again Spotify: being a social website, each track is typically annotated with genre, artist, tags and other related information by many users of the application. Moreover, data from several websites may be easily retrieved and aggregated using the provided programming interfaces. Overall, this amount in an extremely large mass of information on which efficient data mining is challenging. These reasons are making AMC tasks an interesting benchmark for machine learning tools. For example the MIREX challenge [2] has seen a constant growth over the last years, and today comprises more that fifteen different tasks regarding AMC.

In this paper we test Extreme Learning Machines (ELM) [3] on several audio-related benchmarks. ELM is a relatively new learning technique that we believe of great interest for audio classification. In particular, ELM models are highly versatile (providing a unified solution for both multi-class classification and regression), and are much faster to train than standard models such as neural networks. The main idea of ELM is projecting the original input into an highly dimensional feature space, where a linear model is subsequently applied. The peculiarity is that this new space is fully fixed before observing the data, hence the actual learning consist of a simple linear regression that can be computed efficiently in closed form.

At this time, we are aware of only two works that have used ELM for music classification. In [4], ELM is applied to the problem of genre classification, on a author-generated dataset. Out of nine tests, ELM has a greater average accuracy than a standard Support Vector Machine. Then, in [5] the authors tested ELM for the classification of Han Chinese folk songs, together with a novel musical encoding method they call MFDMap. However, no comparisons are made with other classifiers. Thus, no work has been done up to now

[1] https://www.spotify.com/

to test and compare ELM on standard musical benchmarks, which is the main subject of this paper. In particular, ELM is tested on five publicly available datasets, for a total of six experiments comprising four different AMC tasks (genre classification, mood classification, music/speech discrimination and year recognition). Results are highly promising. As can be expected, in all cases they show a strong decrease in the computational time required for training with respect to a standard neural network model. Moreover, this is obtained whilst scoring higher or comparable results in term of accuracy.

The rest of the paper is organized as follows: in Section II we formulate the problem of automatic music classification. Section III is devoted to a brief overview of ELM theory. Section IV details all the experiments. Then, we conclude on Section V, where we also provide some final remarks.

II. Automatic Music Classification

Automatic classification [6] is the problem of retrieving an unknown relation between an *input space* X and an *output space* Y, where Y contains only finitely many elements. The only information we have is contained in a dataset of N *examples* $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ of the relation, that we call the *training set*. The algorithm is generally tested on a second, independent dataset that we call the *testing set*.

Considering the case of mono-channel encoding, a song can be described by a vector of amplitudes of the form:

$$\mathbf{s} = [s[1], \ldots, s[n]]^T$$

However, this raw format is seldom used for classification purposes due to its high dimensionality and low information content. In general, a d-dimensional vector of features $\mathbf{x} \in \mathbb{R}^d$, is extracted from each song and used as input for the classification step. A non-comprehensive list of possible features includes:

- Spectral features such as *Spectral Centroid*, *Spectral Rolloff* and others.
- Further elaborations of the spectral features, including for example Mel-frequency cepstral coefficients (MFCC) [7].
- Temporal features, typically constructed starting from statistics of different orders of the original signal (mean, variance...).
- Higher-level features, including descriptors of pitch and rhythm.
- Meta-information on the song, such as the genre, the artist or the year of release.
- User-generated tags. This tags may also overlap with the meta-information described before.

We already stated in Section I that the problem of choosing an optimal set of features from those listed above is not trivial. Since we restrict ourselves to standard benchmarks, in this paper we do not discuss further this issue and refer the interested reader to [1], in particular Section II.

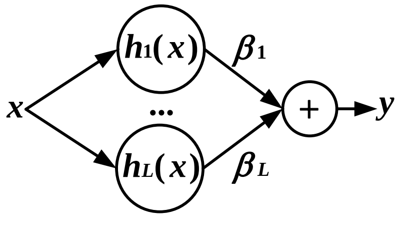

Fig. 1. ELM basic structure.

The output space Y contains all the possible labellings of a song. For example, in a genre classification task Y may be comprised of the labels $\{\text{rock}, \text{pop}, \text{classical}\}$. In some problems an input vector may belong to more than one label (e.g. automatic tagging). In this case, a simple solution is to consider each label as a separate binary classification task.

In the case of M disjoint classes, they can be encoded as an M-dimensional binary vector $\mathbf{y} = \{0, 1\}^M$, with $y_i = 1$ when the input is of class i. This is known as *dummy encoding* of the output and will be used extensively in our experiments.

III. Extreme Learning Machines

An Extreme Learning Machine (ELM) is a model of the form [3]:

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{L} h_i(\mathbf{x})\beta_i = \mathbf{h}^T(\mathbf{x})\boldsymbol{\beta} \tag{1}$$

where $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_L(\mathbf{x})]^T$ is called the *ELM feature vector*, and $\boldsymbol{\beta}$ represents the vector of expansion coefficients. Equation (1) is equivalent to a two-layered network, where the input is first projected to an L-dimensional space, over which a linear combination is performed. This architecture is shown in more detail in Fig. 1.

The ELM feature vector is similar to the hidden layer of a standard feedforward neural network. It is, however, fully fixed in advance, hence it has no free parameters to be tuned. To construct this space, we can perform an operation of *randomization*. Consider a family of functions $g(\mathbf{x}, \boldsymbol{\theta}), \mathbf{x} \in X$, indexed by the parameter vector $\boldsymbol{\theta}$. If we draw $\boldsymbol{\theta}$ L times according to a uniform probability distribution, the resulting functions form exactly a feature space as in (1). The main result of ELM theory (see Huang et al. [3]) is that almost any nonlinear, piecewise continuous function can be used in such a way, granting the resulting network with universal approximation capability.

In this work we consider the original implementation of ELM [8], where the optimal vector of expansion coefficients is found by solving:

$$\text{minimize } \|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|_2^2 \text{ and } \|\boldsymbol{\beta}\|_2 \tag{2}$$

where we defined the hidden matrix $\mathbf{H} = [\mathbf{h}(\mathbf{x}_1), \ldots, \mathbf{h}(\mathbf{x}_N)]$ and the output matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T$. Choosing the solution of minimum norm has several justifications, and has solid theoretical roots in neural network theory [6]. Solution to (2) is found by:

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{Y} \tag{3}$$

where $\mathbf{H}^{\dagger}$ is the *Moore-Penrose* inverse of $\mathbf{H}$. There exists several methods to compute $\mathbf{H}^{\dagger}$. For example, whenever $\mathbf{H}^T\mathbf{H}$ is non-singular, we have $\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$.

Note that a more general formulation exists [3], where the expansion coefficient is found by solving a regularized optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\beta}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{C}{2}\sum_{i=1}^{N}\boldsymbol{\zeta}_i^2 \\ \text{subject to} \quad & \mathbf{h}^T(x_i)\boldsymbol{\beta} = \mathbf{y}_i - \boldsymbol{\zeta}_i,\ i = 1,\ldots,N. \end{aligned} \tag{4}$$

where each $\boldsymbol{\zeta}_i, i = 1,\ldots,N$ measure the error between desired and predicted output. However, for simplicity we do not consider this formulation here. As a final remark, we note that a binary classifier can be easily constructed from ELM by considering the thresholded output:

$$f'(\mathbf{x}) = sign(f(\mathbf{x})) \tag{5}$$

While a multi-class classifier using the dummy encoding described in Section II is constructed as:

$$f'(\mathbf{x}) = \arg\max_{i\in 1,\ldots,M} f_i(\mathbf{x})$$

IV. Experimental Results

We tested ELM on several music-related, publicly available benchmarks. As a baseline algorithm we used a standard feedforward neural network [6]. Optimal parameters for the neural network were found by replicating the experiments in the original papers describing the datasets to obtain comparable results. Parameters for the ELM, instead, were found by cross-validating on an independent portion of the dataset. All simulations were performed by MATLAB 2012a, on an Intel i3 3.07 GHz processor at 64 bit, with 4 GB of RAM available, and each result is averaged over 100 runs. The neural network is constructed using the Neural Network Toolbox of Matlab and trained using standard gradient descent backpropagation. A sigmoid activation function was used to construct the ELM feature space:

$$g(\mathbf{a},\mathbf{x},b) = \frac{1}{1 + e^{-(\mathbf{ax}+b)}} \tag{6}$$

All the experiments are summarized in Table I, where the accuracy is provided together with the average training time in brackets (except for the fourth experiment where the average root mean-square error is provided). Results are consistent between each experiment, showing that ELM achieves a comparable or higher accuracy than the standard neural network but is extremely faster in training. Each experiment is detailed in the following.

A. Music/Speech Discrimination

The first experiment is on automatic discrimination of music/speech, taken from the popular dataset GTZAN [9]. It consists of 120 tracks, each 30 seconds long, equally subdivided in the two classes. Tracks are 16-bit audio files in .wav format encoded in mono. As input vector we used a set of 13 Mel-frequency cepstral coefficients (MFCC) extracted using the MIR Toolbox[2]. The choice is a standard one, since MFCC coefficients are known to perform well on speech discrimination.

Data is randomly split into 66% for the training set and the remaining 34% for the testing set. As can be seen from the first row of Table I, ELM has a marginal advantage in terms of accuracy, but is two orders of magnitude faster in training.

B. Genre Classification (Dortmund)

In a second experiment, we considered the problem of genre classification. The dataset used here is the Dortmund dataset (also known as garageband) [10]. The input to the classifier consists of 49 features extracted according to the search method detailed in [11]. There is a total of 1886 songs subdivided into 9 possible classes (alternative, blues, electronic, folkcountry, funksoulrnb, jazz, pop, raphiphop and rock). The songs were randomly split into 80% for training and 20% for testing. Missing values were replaced with the average value for the attribute relative to all the other examples in the class.

The second row of Table I shows the results. ELM obtains a far more efficient classification accuracy with respect to the neural network (54 % instead of 48 %). Moreover, it took in average half a second to train compared to the 3 seconds and half for the neural network.

C. Genre Classification (LMD)

The third experiment is again of genre classification, but with an harder dataset. In particular, we used the Latin Music Database [12]. It consists of features extracted from 3160 music pieces belonging to one of ten possible classes: Tango, Bolero, Batchata, Salsa, Merengue, Ax, Forr, Sertaneja, Gacha and Pagode. The features are extracted from the 30 seconds in the middle of each piece. They are divided into three groups: Timbral Texture, Beat Related and Pitch Related (see the original paper and [9] for more details).

We used the same 10 splits as in the original paper and report our results in the third row of Table I. Here the classification accuracy is equivalent in the two cases (60 %). Again, the difference in training time is always of two orders of magnitude.

D. Year/Decade Recognition

The fourth and fifth experiments are a non conventional task, namely year recognition of a song. The dataset that we used is the YearPredictionMSD from the UCI repository[3], which is itself a subset of the Million Song Dataset [13].

[2] https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox

[3] http://archive.ics.uci.edu/ml/datasets/YearPredictionMSD

Experiment	Neural Net	ELM
Music/Speech (GTZAN)	75 % (0.38 sec.)	76 % (0.0076 sec.)
Genre Recognition (DORTMUND)	48 % (3.59 sec.)	54 % (0.54 sec.)
Genre Recognition (LMD)	60 % (16.05 sec.)	60 % (0.27 sec.)
Year Recognition (UCI)	2.6 (1099.91 sec.)	2.6 (11.51 sec.)
Decade Recognition (UCI)	61 % (45.74 sec.)	62 % (9.95 sec.)
Mood Recognition (CAL500)	76 % (0.85 sec.)	75 % (0.011 sec.)

TABLE I. RESULTS OF THE EXPERIMENTS.

The input is composed of 90 attributes regarding the timbre extracted from each song. The output is the release year of the song ranging from 1922 to 2011.

In a first experiment, to simplify the problem, we considered only songs ranging from 2000 to 2011. We selected the first 10000 for training and the last 1000 for testing. Differently from all the other experiments, we treat the problem as a regression one (i.e., with a single continous valued output). In the fourth row of Table I we show the average root-mean square error (RMSE) computed as:

$$RMSE(S) = \sqrt{\frac{1}{|S|}\sum_{i=1}^{|S|}(y_i - f(\mathbf{x}_i))^2} \qquad (7)$$

Performance are similar, but the difference in training time is notable (several minutes against ten seconds). Note, however, that the RMSE remains high, with an average error of more than two years on the prediction.

In the fifth experiment we eased the task by considering only the release decade of the song, and results are presented in the fifth row of Table I. Here the task was treated again as a classification task with a dummy encoding on the decade. Results are comparable with the others. With respect to the previous experiment, however, backpropagation converges faster here.

E. Mood Recognition

The last experiment is on mood recognition. The dataset we used is the CAL500 dataset [14], composed of 500 songs of western music. Each song is annotated with several tags taken from 135 musical-related concepts, including genre, vocal characteristics and mood. Tags were assigned by several students, and then chosen on the basis of a majority vote.

As input to our system we considered the mean value of the 59 Dynamic MFCC features present in the dataset. We then tested ELM on 36 different binary classification tasks, one for each tag associated to a mood. There are a total of 18 possible moods, and for each one its corresponding negation is also present. For example the first two tags are *Angry/Agressive* and *NOT Angry/Agressive*.

The results averaged over the 36 tasks are presented on the last row of Table I. They are in line with the rest, with a comparable accuracy and a significant lower training time for the ELM model.

V. CONCLUSIONS

We tested an Extreme Learning Machine (ELM) on several benchmarks related to audio classification problems. In all of them, the ELM performed well with respect to a standard feedforward neural network, achieving a higher or comparable accuracy with a significantly faster training time.

It is clear that a highly specialized classifier built on a neural network (for example using Gaussian Mixture Models or ensemble of classifiers) may outperform our results on each of the dataset that we considered. However, due to our results, we believe that ELM can provide a good basis for crafting new models to perform automatic music classification. To this end, we plan to provide in a future work a comparison of a specialized ELM on the full Million Song Dataset database [13].

REFERENCES

[1] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A Survey of Audio-Based Music Classification and Annotation," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 303–319, Apr. 2011.

[2] J. S. Downie, "The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research," *Acoustical Science and Technology*, vol. 29, no. 4, pp. 247–255, 2008.

[3] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 2, pp. 513–29, 2012.

[4] Q.-J. B. Loh and S. Emmanuel, "ELM for the Classification of Music Genres," in *2006 9th International Conference on Control, Automation, Robotics and Vision*. Ieee, 2006, pp. 1–6.

[5] S. Khoo, Z. Man, and Z. Cao, "Automatic han chinese folk song classification using extreme learning machines," in *AI 2012: Advances in Artificial Intelligence*, 2012, pp. 49–60.

[6] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, 2nd ed., 2009.

[7] B. Logan *et al.*, "Mel frequency cepstral coefficients for music modeling," in *International Symposium on Music Information Retrieval*, vol. 28, 2000, p. 5.

[8] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, p. 489501, Dec 2006.

[9] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, p. 293302, 2002.

[10] K. Morik and M. Wurst, “A benchmark dataset for audio classification and clustering,” in *Proceedings of the international conference on music information retrieval*.

[11] I. Mierswa and K. Morik, “Automatic feature extraction for classifying audio data,” *Machine Learning*, pp. 1–28, 2005.

[12] C. N. Silla, C. A. Kaestner, and A. L. Koerich, “Automatic music genre classification using ensemble of classifiers,” in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*. IEEE, 2007, pp. 1687–1692.

[13] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[14] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, “Semantic annotation and retrieval of music and sound effects,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 2, pp. 467–476, February 2008.