Comparing Shallow versus Deep Neural Network Architectures for Automatic Music Genre Classification

Alexander Schindler
Austrian Institute of Technology
Digital Safety and Security
Vienna, Austria
alexander.schindler@ait.ac.at

Thomas Lidy, Andreas Rauber
Vienna University of Technology
Institute of Software Technology
Vienna, Austria
lidy,rauber@ifs.tuwien.ac.at

Abstract

In this paper we investigate performance differences of different neural network architectures on the task of automatic music genre classification. Comparative evaluations on four well known datasets of different sizes were performed including the application of two audio data augmentation methods. The results show that shallow network architectures are better suited for small datasets than deeper models, which could be relevant for experiments and applications which rely on small datasets. A noticeable advantage was observed through the application of data augmentation using deep models. A final comparison with previous evaluations on the same datasets shows that the presented neural network based approaches already outperform state-of-the-art handcrafted music features.

1 Introduction

Music classification is a well researched topic in Music Information Retrieval (MIR) [FLTZ11]. Generally, its aim is to assign one or multiple labels to a sequence or an entire audio file, which is commonly accomplished in two major steps. First, semantically meaningful audio content descriptors are extracted from the sampled audio signal. Second, a machine learning algorithm is applied, which attempts to discriminate between the classes by finding separating boundaries in the multidimensional feature-spaces. Especially the first step requires extensive knowledge and skills in various specific research areas such as audio signal processing, acoustics and/or music theory. Recently many approaches to MIR problems have been inspired by the remarkable success of Deep Neural Networks (DNN) in the domains of computer vision [KSH12], where deep learning based approaches have already become the *de facto standard*. The major advantage of DNNs are their *feature learning* capability, which alleviates the domain knowledge and time intensive task of crafting audio features by hand. Predictions are also made directly on the modeled input representations, which is commonly raw input data such as images, text or audio spectrograms. Recent accomplishments in applying Convolutional Neural Networks (CNN) to audio classification tasks have shown promising results by outperforming conventional approaches in different evaluation campaigns such as the Detection and Classification of Acoustic Scenes and Events (DCASE) [LS16a] and the Music Information Retrieval Evaluation EXchange (MIREX) [LS16b].

An often mentioned paradigm concerning neural networks is that deeper networks are better in modeling non-linear relationships of given tasks [SLJ+15]. So far preceding MIR experiments and approaches reported in literature have not explicitly demonstrated the advantage of deep over shallow network architectures in a magnitude similar to results reported from the computer vision domain. This may be related to the absence of similarly large datasets as they are available in the visual related research areas. A special focus of this paper is thus set on the performance of neural networks on small datasets, since data availability is still a problem in MIR, but also because many tasks involve the processing of small collections. In this paper we present a performance evaluation of shallow and deep neural network architectures. These models and the applied method will be detailed in Section 2. The evaluation will be performed on well known music genre classification datasets in the domain of Music Information Retrieval. These datasets and the evaluation procedure will be described in Section 3. Finally we draw conclusions from the results in Section 5 and give an outlook to future work.

In: W. Aigner, G. Schmiedl, K. Blumenstein, M. Zeppelzauer (eds.): Proceedings of the 9th Forum Media Technology 2016, St. Pölten, Austria, 24-11-2016, published at http://ceur-ws.org

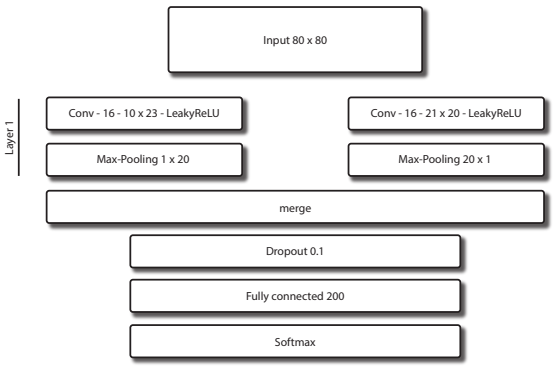

Figure 1: Shallow CNN architecture

2 Method

The parallel architectures of the neural networks used in the evaluation are based on the idea of using a time and a frequency pipeline described in [PLS16], which was successfully applied in two evaluation campaigns [LS16a, LS16b]. The system is based on a parallel CNN architecture where separate CNN Layers are optimized for processing and recognizing music relations in the frequency domain and to capture temporal relations (see Figure 1).

The Shallow Architecture: In our adaption of the CNN architecture described in [PLS16] we use two similar pipelines of CNN Layers with 16 filter kernels each followed by a Max Pooling layer (see Figure 1). The left pipeline aims at capturing frequency relations using filter kernel sizes of 10×23 and Max Pooling sizes of 1×20. The resulting 16 vertical rectangular shaped feature map responses of shape 80×4 are intended to capture spectral characteristics of a segment and to reduce the temporal complexity to 4 discrete intervals. The right pipeline uses a filter of size 21×20 and Max Pooling sizes of 20×1. This results in horizontal rectangular shaped feature maps of shape 4×80. This captures temporal changes in intensity levels of four discrete spectral intervals. The 16 feature maps of each pipeline are flattened to a shape of 1×5120 and merged by concatenation into the shape of 1×10240, which serves as input to a 200 units fully connected layer with a dropout of 10%.

The Deep Architecture: This architecture follows the same principles of the shallow approach. It uses a parallel arrangement of rectangular shaped filters and Max-Pooling windows to capture frequency and temporal relationships at once. But, instead of using the information of the large feature map responses, this architecture applies additional CNN and pooling layer pairs (see Figure 2). Thus, more units can be applied to train on the subsequent smaller input feature maps. The first level of the parallel layers are similar to the original approach. They use filter kernel sizes of 10×23 and 21×10 to capture frequency and temporal relationships. To retain these characteristics the sizes of the convolutional filter kernels as well as the feature maps are sub-sequentially divided in halves by the second and third layers. The filter and Max Pooling sizes of the fourth layer

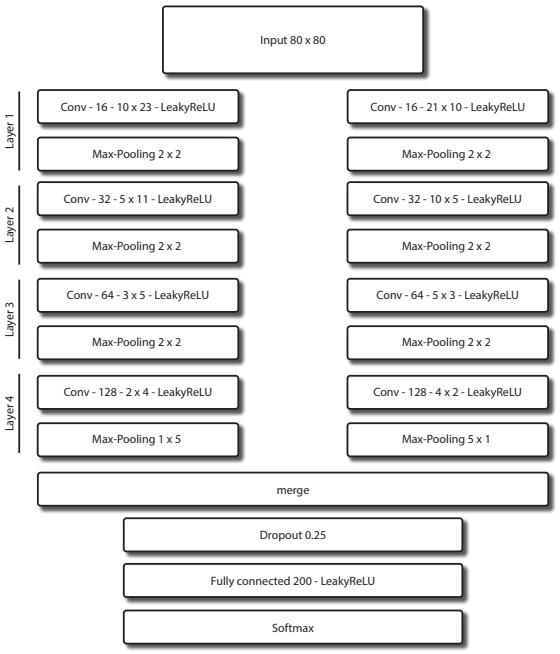

Figure 2: Deep CNN architecture

are slightly adapted to have the same rectangular shapes with one part being rotated by 90°. As in the shallow architecture the same sizes of the final feature maps of the parallel model paths balances their influences on the following fully connected layer with 200 units with a 25% dropout rate.

2.1 Training and Predicting Results

In each epoch during training the network multiple training examples sampled from the segment-wise log-transformed Mel-spectrogram analysis of all files in the training set are presented to both pipelines of the neural network. Each of the parallel pipelines of the architectures uses the same 80 × 80 log-transformed Mel-spectrogram segments as input. These segments have been calculated from a fast Fourier transformed spectrogram using a window size of 1024 samples and an overlap of 50% from 0.93 seconds of audio transformed subsequently into Mel scale and Log scale.. For each song of a dataset 15 segments have been randomly chosen.

All trainable layers used the *Leaky ReLU* activation function [MHN13], which is an extension to the ReLU (Rectifier Linear Unit) that does not completely cut off activation for negative values, but allows for negative values close to zero to pass through. It is defined by adding a coefficient α in $f(x) = \alpha x$, for $x < 0$, while keeping $f(x) = x$, for $x \geq 0$ as for the ReLU. In our architectures, we apply Leaky ReLU activation with $\alpha = 0.3$. L_1 weight regularization with a penalty of 0.0001 was applied to all trainable parameters. All networks were trained towards *categorical-crossentropy* objective using the stochastic *Adam* optimization [KB14] with $beta_1 = 0.9$, $beta_2 = 0.999$, $epsilon = 1e-08$ and a learning rate of 0.00005.

The system is implemented in Python and using *librosa*

[MRL+15] for audio processing and Mel-log-transforms and *Theano*-based library *Keras* for Deep Learning.

2.1.1 Data Augmentation

To increase the number of training instances we experiment with two different audio data augmentation methods. The deformations were applied directly to the audio signal preceding any further feature calculation procedure described in Section 2.1. The following two methods were applied using the MUDA framework for musical data augmentation [MHB15]:

Time Stretching: slowing down or speeding up the original audio sample while keeping the same pitch information. Time stretching was applied using the multiplication factors 0.5, 0.2 for slowing down and 1.2, 1.5 for increasing the tempo.

Pitch Shifting: raising or lowering the pitch of an audio sample while keeping the tempo unchanged. The applied pitch shifting lowered and raised the pitch by 2 and 5 semitones.

For each deformation three segments have been randomly chosen from the audio content. The combinations of the two deformations with four different factors each resulted thus in 48 additional data instances per audio file.

3 Evaluation

As our system analyzes and predicts multiple audio segments per input file, there are several ways to perform the final prediction of an input instance:

Raw Probability: The raw accuracy of predicting the segments as separated instances ignoring their file dependencies.

Maximum Probability: The output probabilities of the Softmax layer for the corresponding number of classes of the datasets are summed up for all segments belonging to the same input file. The predicted class is determined by the maximum probability among the classes from the summed probabilities.

Majority Vote: Here, the predictions are made for each segment processed from the audio file as input instance to the network. The class of an audio segment is determined by the maximum probability as output by the Softmax layer for this segment instance. Then, a majority vote is taken on all predicted classes from all segments of the same input file. Majority vote determines the class that occurs most often.

We used stratified 4-fold cross validation. Multi-level stratification was applied paying special attention to the multiple segments used per file. It was ensured that the files were distributed according their genre distributions and that no segments of a training file was provided in the corresponding test split.

The experiments were grouped according to the four different datasets. For each dataset the performances for the shallow and deep architecture were evaluated followed by the experiments including data augmentation. The architectures were further evaluated according their performance after a different number of training epochs. The networks were trained and evaluated after 100 and 200 epochs without early stopping. Preceding experiments showed that test accuracy could improve despite rising validation loss though on smaller sets no significant improvement was recognizable after 200 epochs. For the experiments with data augmentation, the augmented data was only used to train the networks (see Table 3.1). For testing the network the original segments without deformations were used.

			Train		Test
Data	**Tracks**	**cls**	**wo. au.**	**w. au.**	
GTZAN	1,000	10	11,250	47,250	3,750
ISMIR G.	1,458	6	16,380	68,796	5,490
Latin	3,227	10	36,240	152,208	12,165
MSD	49,900	15	564,165	—	185,685

Table 1: Overview of the evaluation datasets, their number of classes (cls) and their corresponding number of test and training data instances without (wo. au.) and with (w. au.) data augmentation.

3.1 Data Sets

For the evaluation four data sets have been used. We have chosen these datasets due to their increasing number of tracks and because they are well known and extensively evaluated in the automatic genre classification task. This should also provide comparability with experiments reported in literature.

GTZAN: This data set was compiled by George Tzanetakis [Tza02] in 2000-2001 and consists of 1000 audio tracks equally distributed over the 10 music genres: blues, classical, country, disco, hiphop, pop, jazz, metal, reggae, and rock.

ISMIR Genre: This data set has been assembled for training and development in the ISMIR 2004 Genre Classification contest [CGG+06]. It contains 1458 full length audio recordings from Magnatune.com distributed across the 6 genre classes: Classical, Electronic, JazzBlues, MetalPunk, RockPop, World.

Latin Music Database (LMD): [SKK08] contains 3227 songs, categorized into the 10 Latin music genres Axé, Bachata, Bolero, Forró, Gaúcha, Merengue, Pagode, Salsa, Sertaneja and Tango.

Million Song Dataset (MSD): [BMEWL11] a collection of one million music pieces, enables methods for large-scale applications. It comes as a collection of metadata such as the song names, artists and albums, together with a set of features extracted with the The Echo Nest services, such as loudness, tempo, and MFCC-like features. We used the CD2C genre assignments as ground truth [Sch15] which are an adaptation of the MSD genre label assignments presented in [SMR12]. For the experiments a sub-set of approximately 50.000 tracks was sub-sampled.

4 Results

The results of the experiments are provided in Table 4. For each dataset all combinations of experimental results were tested for significant difference using a Wilcoxon signed-rank test. None of the presented results showed a significant difference for $p<0.05$. Thus, we tested at the next higher level $p<0.1$. The following observations on the datasets were made:

GTZAN: Training the models with 200 epochs instead of only 100 epochs significantly improved the *raw* and *max* accuracies for the shallow models. An additional test on training 500 epochs showed no further increase in accuracy for any of the three prediction methods. Training longer had no effect on the deep model due to early over-fitting. No significant differences were observed between shallow and deep models except for the raw prediction values of the shallow model (200 epochs) exceeding those of the deep model (200 epochs). While the improvements through data augmentation on deep models compared to the un-augmented longer trained deep models are not significant, considerable improvements of 4.2% were observed for models trained for the same number of epochs. An interesting observation is the negative effect of data augmentation on the shallow models where longer training outperformed augmentation.

ISMIR Genre: Training more epochs only had a significant positive effect on the *max* and *maj* values of the deep model but none for the shallow ones. The deep models showed no significant advantage over the shallow architectures which also showed higher *raw* prediction values even on shorter trained models. Data augmentation improved the predictions of both architectures with significant improvements for the *raw* values. Especially the deep models significantly profited from data augmentation with *max* values increased by 3.08% for models trained for the same number of epochs and 2.05% for the longer trained models. The improvements from deep over shallow models using augmented data were only significant for the *raw* values.

Latin: Training more epochs only had a positive effect for the *raw* and *max* values of the shallow model, but not for the deep architecture. On this dataset, the deep model significantly outperformed the shallow architecture including the shallow model trained using data augmentation. Data augmentation improved the significantly improved the performance of the deep models by 1,61% for the *max* values. Similar to the GTZAN dataset, data augmentation showed a degrading effect on the shallow model which showed significantly higher accuracy values by training for more epochs.

MSD: A not significant advantage of deep over shallow models was observed. Experiments using data augmentation and longer training were omitted due to the already large variance provided by the MSD which multiplies the preceding datasets by factors from 15 to 50.

D	Model	raw	max	maj	ep
GTZAN	shallow	66.56 (0.69)	78.10 (0.89)	77.80 (0.52)	100
	deep	65.69 (1.23)	78.60 (1.97)	78.00 (2.87)	100
	shallow	67.49 (0.39)	80.80 (1.67)	80.20 (1.68)	200
	deep	66.19 (0.55)	80.60 (2.93)	80.30 (2.87)	200
	shallow aug	66.77 (0.78)	78.90 (2.64)	77.10 (1.19)	100
	deep aug	**68.31** (2.68)	**81.80** (2.95)	**82.20** (2.30)	100
ISMIR Genre	shallow	75.66 (1.30)	85.46 (1.87)	84.77 (1.43)	100
	deep	74.53 (0.52)	84.08 (1.13)	83.95 (0.97)	100
	shallow	75.43 (0.65)	84.91 (1.96)	85.18 (1.27)	200
	deep	74.51 (1.71)	85.12 (0.76)	85.18 (1.23)	200
	shallow aug	76.61 (1.04)	86.90 (0.23)	86.00 (0.54)	100
	deep aug	**77.20** (1.14)	**87.17** (1.17)	**86.75** (1.41)	100
Latin	shallow	79.80 (0.95)	92.44 (0.86)	92.10 (0.97)	100
	deep	81.13 (0.64)	94.42 (1.04)	94.30 (0.81)	100
	shallow	80.64 (0.83)	93.46 (1.13)	92.68 (0.88)	200
	deep	81.06 (0.51)	95.14 (0.40)	94.83 (0.53)	200
	shallow aug	78.09 (0.68)	92.78 (0.88)	92.03 (0.81)	100
	deep aug	**83.22** (0.83)	**96.03** (0.56)	**95.60** (0.58)	100
MSD	shallow	58.20 (0.49)	63.89 (0.81)	63.11 (0.74)	100
	deep	**60.60** (0.28)	**67.16** (0.64)	**66.41** (0.52)	100

Table 2: Experimental results for the evaluation datasets (D) at different number of training epochs (ep): Mean accuracies and standard deviations of the 4-fold cross-evaluation runs calculated using raw prediction scores (raw) and the file based maximum probability (max) and majority vote approach (maj).

5 Conclusions and Future Work

In this paper we evaluated shallow and deep CNN architectures towards their performance on different dataset sizes in music genre classification tasks. Our observations showed that for smaller datasets shallow models seem to be more appropriate since deeper models showed no significant improvement. Deeper models performed slightly better in the presence of larger datasets, but a clear conclusion that deeper models are generally better could not be drawn. Data augmentation using time stretching and pitch shifting significantly improved the performance of deep models. For shallow models on the contrary it showed a negative effect on the small datasets. Thus, deeper models should be considered when applying data augmentation. Comparing the presented results with previously reported evaluations on the same datasets [SR12] shows, that the CNN based approaches already outperform handcrafted music features such as the Rhythm Patterns (RP) family [LSC+10] (highest values: GTZAN 73.2%, ISMIR Genre 80.9%, Latin 87.3%) or the in the referred study presented Temporal Echonest Features [SR12] (highest values: GTZAN 66.9%, ISMIR Genre 81.3%, Latin 89.0%).

Future work will focus on further data augmentation methods to improve the performance of neural networks on small datasets and the Million Song Dataset as well as on different network architectures.

References

[BMEWL11] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *ISMIR*, volume 2, page 10, 2011.

[CGG+06] Pedro Cano, Emilia Gómez, Fabien Gouyon, Perfecto Herrera, Markus Koppenberger, Beesuan Ong, Xavier Serra, Sebastian Streich, and Nicolas Wack. ISMIR 2004 audio description contest. Technical report, 2006.

[FLTZ11] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *Multimedia, IEEE Transactions on*, 13(2):303–319, 2011.

[KB14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[LS16a] Thomas Lidy and Alexander Schindler. CQT-based convolutional neural networks for audio scene classification. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, pages 60–64, September 2016.

[LS16b] Thomas Lidy and Alexander Schindler. Parallel convolutional neural networks for music genre and mood classification. Technical report, Music Information Retrieval Evaluation eXchange (MIREX 2016), August 2016.

[LSC+10] Thomas Lidy, Carlos N. Silla, Olmo Cornelis, Fabien Gouyon, Andreas Rauber, Celso A. A. Kaestner, and Alessandro L. Koerich. On the suitability of state-of-the-art music information retrieval methods for analyzing, categorizing, structuring and accessing non-western and ethnic music collections. *Signal Processing*, 90(4):1032–1048, 2010.

[MHB15] Brian McFee, Eric J Humphrey, and Juan P Bello. A software framework for musical data augmentation. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[MHN13] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. *ICML 2013*, 28, 2013.

[MRL+15] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th Python in Science Conference*, 2015.

[PLS16] Jordi Pons, Thomas Lidy, and Xavier Serra. Experimenting with musically motivated convolutional neural networks. In *Proceedings of the 14th International Workshop on Content-based Multimedia Indexing (CBMI 2016)*, Bucharest, Romania, June 2016.

[Sch15] Hendrik Schreiber. Improving genre annotations for the million song dataset. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2012)*, Malaga, Spain, 2015.

[SKK08] C N Silla Jr., Celso A A Kaestner, and Alessandro L Koerich. The Latin Music Database. In *Proceedings of the 9th International Conference on Music Information Retrieval*, pages 451—-456, 2008.

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[SMR12] Alexander Schindler, Rudolf Mayer, and Andreas Rauber. Facilitating comprehensive benchmarking experiments on the million song dataset. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, pages 469–474, Porto, Portugal, October 8-12 2012.

[SR12] Alexander Schindler and Andreas Rauber. Capturing the temporal domain in echonest features for improved classification effectiveness. In *Adaptive Multimedia Retrieval*, Lecture Notes in Computer Science, Copenhagen, Denmark, October 24-25 2012. Springer.

[Tza02] G. Tzanetakis. *Manipulation, analysis and retrieval systems for audio signals*. PhD thesis, 2002.