PROGRESSIVE TRAINING OF CONVOLUTIONAL NEURAL NETWORKS FOR ACOUSTIC EVENTS CLASSIFICATION

Federico Colangelo, Federica Battisti, Alessandro Neri

Università degli studi Roma Tre
Rome, Italy
federico.colangelo, federica.battisti, alessandro.neri@uniroma3.it

ABSTRACT

Convolutional neural networks represent the state of the art in multiple fields. Techniques that improve the training of these models are of prime interest since they have the capability to improve performances on a large variety of tasks. In this paper, we investigate the performance of progressive resizing, originally introduced in computer vision, when applied to the training of convolutional neural networks for audio events classification. We evaluate the original resizing algorithm and introduce a novel one, comparing the performances against a baseline system. Two of the most relevant audio datasets are used for assessing the performances of the proposed approach. Experimental results suggest that progressive resizing methods improves the performances of audio events classification models. The novel approach introduces a complimentary gain in performances with respect to the original technique.

Index Terms— Deep learning, acoustic events, classification, tagging

1. INTRODUCTION

Deep learning models represent the state-of-the-art in most multimedia applications. The performances of these models have experienced tremendous advancements in the last years, enabled by the availability of data, and improved hardware and techniques, such as network architectures and training procedures.

Deep learning models for acoustic events classification have been increasingly adopting 2D signals' specific techniques originally developed in the field of computer vision. This trend is noticeable in the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge [1]: in the past few years, all the best performing models for audio event classification tasks were variants of a Convolutional Neural Network (CNN) originally designed for images (e.g. ResNet [2], VGGnet [3]), often with minimal adaptation [4, 5]. This is mainly due to two factors. First, CNNs for image classification have increased their performances and have introduced architectural changes that improve training independently from the nature of the 2D signal. For example, ResNets have been shown to be easier to optimize [6]. Second, the use of time-frequency representations for audio, such as the mel-spectrogram, allows to process audio as a 2D signal for the purpose of CNN training. Data augmentation techniques based on this principle have been proposed, obtaining considerable improvements in performances. Moreover, prior work shows that, by taking into account the differences between spectrograms and natural images in terms of attributes (e.g., smoothness and invariance to translation), further improvement in performances can be achieved [7]. Progressive training of CNN models for image classification has been introduced in [8]. In this approach, the CNN training is split into multiple phases. In the first, the model is trained with small images, obtained by resizing the original dataset. Subsequent phases leverage larger and larger images, until the last phase where the model is trained with the original images. It appears that this approach improves the scale invariance of the representation learned by the network, thus leading to improved performances.

This approach has been successfully applied in other contexts. In [9], the authors train a Generative Adversarial Network (GAN) by progressively feeding images with higher resolution while increasing the number of layers. This procedure allows to stabilize the GAN training enough to generate images with a double resolution with respect to the state-of-the-art.

In [10] the authors propose the use of a CNN for super-resolution. Models performing $3\times$ and $4\times$ super-resolution are initialized with parameters from a $2\times$ network, yielding improved performances and faster convergence with respect to the model's baseline.

Considering the encouraging prior works, in this paper we applied these training techniques to audio data and designed a novel variant. The rationale behind this choice is that spectrograms can be extracted with different resolution in time

The authors would like to thank the Open Innovation Lab for making available the hardware infrastructure located in the Laurentina premises of the Leonardo S.p.A. company to perform the experiments presented in this paper.

and frequency, thus leading to different size of the data used to train the network. With this method, the training data gets progressively more detailed as the training proceeds, due to the increasing resolution of the spectrogram.
In this paper, we make the following contributions:

- the application of progressive resizing based on bi-linear interpolation for training CNNs in acoustic events classification tasks;
- a novel technique for performing progressive resizing based on time-cropping and mel-spectrogram sub-sampling.

The effectiveness of these techniques is evaluated on multiple datasets and compared with a baseline CNN. The rest of this paper is organized as follows: Section 2 describes the proposed methods for progressive training as well as the training of the baseline architecture used for comparison. Section 3 details the implementation of the proposed approach, describes the performed experiments and discusses the obtained results. Finally, in Section 4 the conclusions are drawn.

2. PROPOSED METHOD

The basic concept of progressive resizing is to train the CNN with samples of progressively higher resolution. More specifically, given two networks, one trained with the best single-step configuration and the other by progressively converging to the same configuration, we conjecture that the second training paradigm will perform better.
In order to test this hypothesis, we perform a comparison between the models trained with the two resizing strategies and a baseline setup for audio event classification, as described in the following. First, the input data are re-sampled at 22 kHz. This operation is needed to handle datasets composed by files of varying sampling rates. Silence is cropped from the beginning and the end of each clip. The mel-spectrogram of the signal is then extracted and normalized using min-max normalization so that all its values are in the range of $[0\ 1]$. Mel-spectrograms corresponding to a duration of t seconds are extracted. For audio files longer than t, the spectrogram is extracted and cropped. Shorter audio files are handled by replicating the spectrogram until temporal slices of length t are obtained. The mel-spectrograms are then resized. Two different methods for obtaining resized spectrogram are compared. The first one uses bi-linear interpolation: for each sample in the dataset, the mel-spectrogram is calculated and resized at three different resolutions $R_1 < R_2 < R_3$, being R_3 the full resolution, without resizing. This method represents progressive resizing as introduced in [8] and has been shown to improve the performances in image classification problems.
The second method resizes the spectrograms by cropping a portion of the input in the time dimension and leveraging a mel filterbank with fewer triangular filters. As in the first method, for each sample in the dataset, three versions at resolution R_1, R_2 and R_3 are obtained. Figure 1 shows the effects of the two resizing methods on the spectrograms. It can be noted that spectrograms extracted using fewer mel filters ((b) and (d)) generate a low-pass effect on the final image, thus resulting in the presence of blur in the finer details. We speculate that this effect can yield improvements in the domain of audio classification, as it allows the network to first learn coarser features on the down-sampled spectrogram, and then learn finer features as the training data are progressively sharpened. A final set of spectrograms is extracted with resolution R_3. These spectrograms provide a baseline against which the effects introduced by the resizing procedures are

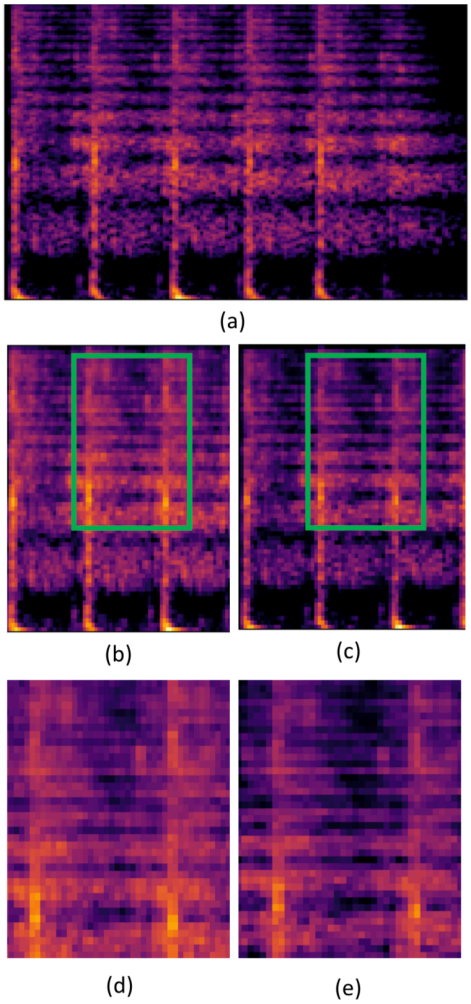

Fig. 1. Comparison of spectrograms: (a) spectrogram at full resolution R_3, (b) and (c) spectrograms at resolution R_1 calculated with the mel method and bi-linear resizing, respectively, (d) and (e) magnified versions of (b) and (c), respectively.

benchmarked. The spectrograms are then used to train a CNN. The selected CNN is a ResNet34 model. It should be noted that no extra steps are taken to improve the model performances (e.g., data augmentation, mix up regularization). This choice stems from the fact that our purpose is to evaluate progressive training in a reference setup.

3. EXPERIMENTAL VALIDATION

In this section, the technical details of the procedure used for training the models are presented. Each model is initialized with parameters learned for the ImageNet [11] dataset. In our experiments, this initialization allowed shorter training duration (i.e. faster convergence), albeit it did not have an impact on the final performances. Based on the assumption that using different Learning Rate (LR)s allows the model to improve its performances, each model is trained in stages. To this purpose different values of LR λ are used for each stage. Before the beginning of each step the best performing model, based on the performances on the held-out set, is loaded. All models are trained with weight decay, with a factor of 0.1. The baseline CNN is trained using data at full resolution R_3 as follows:

- train the network with λ_{max} λ_1 for 20 epochs;
- train the last layer of the network with λ_{max} λ_2 and the other layers with $\frac{\lambda_2}{3}$ for 50 epochs;
- train the last layer of the network with λ_{max} λ_3 and the other layers with $\frac{\lambda_3}{3}$ for 120 epochs.

The networks that use progressive resizing are trained using progressively increasing resolutions as follows:

- using data at the lowest resolution R_1:
  - train the network with λ_{max} λ_{1R1} for 20 epochs;
  - train the network with λ_{max} λ_{2R1} and the other layers with $\frac{\lambda_{2R1}}{3}$ for 20 epochs;
- using data at intermediate resolution R_2:
  - train the network with λ_{max} λ_{1R2} for 20 epochs;
  - train the network with λ_{max} λ_{2R2} and the other layers with $\frac{\lambda_{2R2}}{3}$ for 20 epochs;
- using data at full resolution R_3:
  - train the network with λ_{max} λ_{1R3} for 20 epochs;
  - train the network with λ_{max} λ_{2R3} and the other layers with $\frac{\lambda_{2R3}}{3}$ for 70 epochs.

As can be seen, all the methods use the same number of epochs. Each step follows the *1Cycle* policy introduced in [12]. It consists in varying the LR (λ) and the amount of momentum (m) used to train the model based three phases (i.e. a cycle). In the first phase, λ is initialized with a low value, λ_{min}, and it is increased to the peak value, λ_{max}. Simultaneously, m is initialized at m_{max} and it is decreased to m_{min}. In the second phase, the λ value is decreased from λ_{max} to λ_{min} while the m value is increased from m_{min} to m_{max}. In the third phase, that corresponds to the last n iterations of training, the value of λ decays to zero. The adopted parameters for the *1Cycle* policy were the ones provided by the library implementation [13], except for λ_{max}, which was selected based on the test proposed in [14]. To select an efficient LR, the model is trained with a different λ value for each batch. Each iteration reduces the loss until λ is so large that it causes the loss to diverge. The selected λ is the largest value that achieves a stable reduction in loss. The λ_{max} used for training are reported in Table 1. To validate the proposed method, two state-of-the-art datasets are used: the UrbanSound8K [15] and the ESC-50 [16] datasets.

The UrbanSound8K dataset consists of 8732 short audio clips (i.e. duration <4 seconds), each containing one out of ten acoustic events. The dataset is split into ten folders for cross validation. For each split, 9 folders are used for training and 1 folder is used for testing.

The ESC-50 dataset consists of 2000 5-second long audio clips, each containing one out of 50 acoustic events. The dataset is split into 5 folders for cross-validation. For each split, 4 folders are used for training and 1 folder is used for testing.

For both these sets, the models were trained using the splits defined by the datasets' creators. All spectrograms were extracted using an Short Time Fourier Transform (STFT) window of 1024 samples and an hop size of 256. R_1 mel-spectrograms have a resolution of 64x128. For the mel-based resizing method, this corresponds in using 64 mel filters and t=1.5 seconds. R_2 mel-spectrograms have a resolution of 96x214. For the mel-based resizing method, this leads to the use of 96 mel filters and t=2.5 seconds. R_3 mel-spectrograms have a resolution of 128x300. This corresponds to 128 mel filters and t=3.5 seconds for all setups. Time cropping was performed starting from the beginning of the audio file.

The code was developed using the Fastai [13] and Fastai-

Learning rate	UrbanSound8k	ESC-50	Epochs
λ_1	10^{-2}	$5\text{x}10^{-3}$	20
λ_2	$5\text{x}10^{-6}$	10^{-4}	50
λ_3	10^{-6}	10^{-6}	120
λ_{1R1}	10^{-2}	$5\text{x}10^{-3}$	20
λ_{2R1}	10^{-3}	$5\text{x}10^{-4}$	20
λ_{1R2}	$5\text{x}10^{-3}$	$5\text{x}10^{-3}$	20
λ_{2R2}	$5\text{x}10^{-6}$	$5\text{x}10^{-6}$	20
λ_{1R3}	$5\text{x}10^{-3}$	$5\text{x}10^{-3}$	20
λ_{2R3}	$5\text{x}10^{-6}$	$5\text{x}10^{-6}$	70

Table 1. Values of the learning rates used during training.

audio [17].

Split	Baseline (%)	Mel (%)	Resize (%)
0	79.68	81	84.22
1	79.41	83.57	79.41
2	71.09	71.46	79.15
3	76.98	81.98	75.29
4	73.26	79.7	78.73
5	84.61	85.57	86.43
6	72.52	78.48	75.65
7	66.16	63.45	68.97
8	70.6	73.19	74.88
9	71.59	74.79	71.59
Average	74.59±5.47	77.31±6.66	77.43±5.32

Table 2. Accuracy results for the Urbansound8K dataset

Split	Baseline (%)	Mel (%)	Resize (%)
0	68.5	74.5	71.74
1	81.25	83.5	83.24
2	74	80.25	73.25
3	68.25	73.5	77.49
4	70	72.25	74.25
Average	72.4±5.45	76.8±4.83	75.99±4.56

Table 3. Accuracy results for the ESC-50 dataset.

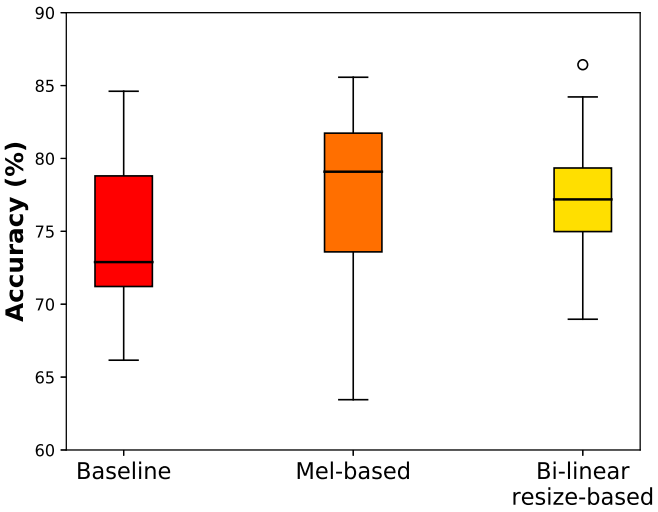

Fig. 2. Box plot of the UrbanSound8K performances.

3.1. Results

Results are shown in Table 2 for the UrbanSound8K dataset and in Table 3 for the ESC-50 dataset. As can be seen, both

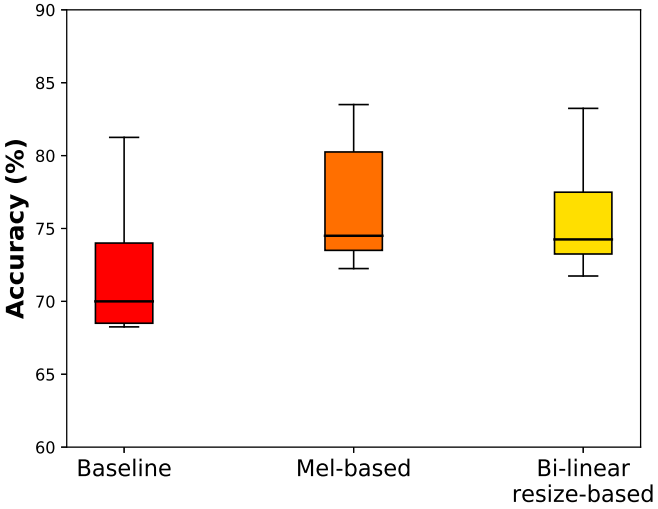

Fig. 3. Box plot of the ESC-50 performances.

methods introduce an improvement with respect to the baseline. These results are far from the current state-of-the-art, which reaches 95.3% accuracy for UrbanSound and 86.4% for ESC [18]. A natural question would be if this improvement would still be present on a state-of-the-art system. While we were not able to include the full investigation in this version of the paper, we found that other augmentation techniques such as Mix-up [19] didn't have any interaction with the proposed method, introducing a flat gain in accuracy with and without progressive resizing. This suggest that the improvement in performances would still be present even on state-of-the-art systems. Looking at the aggregated data, it seems that mel-based resizing has a small advantage over bi-linear resizing. This trend is also highlighted by the box plots shown in Figures 2 and 3. However, the split-by-split results show that the two techniques are complementary. Indeed, while the aggregate statistics are similar, each technique performs best on a subset of the splits. It is thus likely that aggregating the two techniques can yield further gain in performances. The investigation of this approach is left as future work.

4. CONCLUSIONS

CNNs represent the state-of-the-art in audio event classification. In this paper, an evaluation of progressive training of CNN for audio events classification has been presented. A resizing process based on progressively introducing more mel filters has been proposed and compared with the bi-linear resize-based procedure on two of the most important datasets for audio event classification. Experimental results show that both methods significantly improve the performances of the

network, thus suggesting that progressive resizing improves the performance of CNN for audio classification problems. While the effects on performances appear comparable on aggregate, the performances of the two methods are considerably different on the different splits of the dataset. This finding suggests that combining the two techniques can yield further gain in performances. An interesting topic would be to investigate how different data augmentation techniques interact with the proposed technique. Another possible direction is the investigation of the effects of different resizing operators. These ideas will be explored in future work.

5. REFERENCES

[1] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] Tuomas Virtanen, Annamaria Mesaros, Toni Heittola, Aleksandr Diment, Emmanuel Vincent, Emmanouil Benetos, and Benjamin Martinez Elizalde, *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, Tampere University of Technology. Laboratory of Signal Processing, 2017.

[5] Mark D. Plumbley, Christian Kroos, Juan P. Bello, Gaël Richard, Daniel P.W. Ellis, and Annamaria Mesaros, *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, Tampere University of Technology. Laboratory of Signal Processing, 2018.

[6] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein, "Visualizing the loss landscape of neural nets," in *Advances in Neural Information Processing Systems*, 2018, pp. 6389–6399.

[7] Jianfeng Ren, Xudong Jiang, Junsong Yuan, and Nadia Magnenat-Thalmann, "Sound-event classification using robust texture features for robot hearing," *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 447–458, 2016.

[8] Jeremy Howard et al., "Practical deep learning for coders, v3, Lesson 3," "`https://course.fast.ai/`", 2019, "Online; accessed 21-Oct-2019".

[9] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[10] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[12] Leslie N Smith, "A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay," *arXiv preprint arXiv:1803.09820*, 2018.

[13] Jeremy Howard et al., "Fastai," `https://github.com/fastai/fastai`, 2018, "Online; accessed 21-Oct-2019".

[14] Leslie N Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 464–472.

[15] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 1041–1044.

[16] Karol J Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 1015–1018.

[17] "Fastai_ audio," `https://github.com/mogwai/fastai\_audio`, 2019, "Online; accessed 21-Oct-2019".

[18] Yu Su, Ke Zhang, Jingyu Wang, and Kurosh Madani, "Environment sound classification using a two-stream cnn based on decision-level fusion," *Sensors*, vol. 19, no. 7, pp. 1733, 2019.

[19] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.