DMRN+15: DIGITAL MUSIC RESEARCH NETWORK
ONE-DAY WORKSHOP 2020

QUEEN MARY UNIVERSITY OF LONDON
TUE 15 DECEMBER 2020

auraloss: Audio-focused loss functions in PyTorch

Christian J. Steinmetz and Joshua D. Reiss

Centre for Digital Music, Queen Mary University of London, c.j.steinmetz@qmul.ac.uk

***Abstract*— We present `auraloss`[1], a PyTorch package that implements time and frequency domain loss functions designed for audio generation tasks. The package provides a straightforward interface, as well as multichannel support. We demonstrate its application by using each loss function to train a model on the task of emulating an analog dynamic range compressor.**

I. LOSS FUNCTIONS

Error-to-signal ratio — The error-to-signal ratio (ESR) [1] is equivalent to the squared error between the input $\hat{y}$ and target y, both N samples in length, normalized by the energy of the target.

$$\ell_{\text{ESR}}(\hat{y}, y) = \frac{\sum_{i=0}^{N-1} |\hat{y}_i - y_i|^2}{\sum_{i=0}^{N-1} |y_i|^2} \tag{1}$$

Following [2], we also provide perceptually motivated pre-emphasis filters. These include an FIR first-order highpass filter, folded differentiator, as well as an approximation of the A-weighting filter.

Log hyperbolic cosine — The log hyperbolic cosine (log-cosh) [3] aims to strike a balance between the L_1 and L_2. It is similar to the L_2 for small values, providing a level of smoothness, and similar to the L_1 for large values, providing robustness. It is defined in Eq. 2, where a is a hyperparameter that controls the overall smoothness.

$$\ell_{\text{log-cosh}}(\hat{y}, y) = \frac{1}{a} \sum_{i=0}^{N-1} \log(\cosh(a(\hat{y}_i - y_i))) \tag{2}$$

Short-time Fourier transform — The Short-time Fourier transform (STFT) loss is composed of the spectral convergence (Eq. 3), and spectral log-magnitude (Eq. 4), where $||\cdot||_{\text{F}}$ is the Frobenius norm, $||\cdot||_1$ is the L_1 norm, and N is the number of STFT frames. The overall STFT loss is defined as the sum of these two terms [4].

$$\ell_{\text{SC}}(\hat{y}, y) = \frac{\| \, |\text{STFT}(y)| - |\text{STFT}(\hat{y})| \, \|_{\text{F}}}{\| \, |\text{STFT}(y)| \, \|_{\text{F}}} \tag{3}$$

$$\ell_{\text{SM}}(\hat{y}, y) = \frac{1}{N} \left\| \log\left(|\text{STFT}(y)|\right) - \log\left(|\text{STFT}(\hat{y})|\right) \right\|_1 \tag{4}$$

Multi-resolution STFT — The STFT loss can be extended by computing the loss at multiple different resolutions [5]. This improves robustness and avoids potential bias arising from the STFT parameters. The multi-resolution STFT (MR) loss is defined in Eq. 5 as the average of the error at each of the M resolutions.

$$\ell_{\text{MR}}(\hat{y}, y) = \frac{1}{M} \sum_{m=1}^{M} \left(\ell_{\text{SC}}(\hat{y}, y) + \ell_{\text{SM}}(\hat{y}, y)\right). \tag{5}$$

For optimal performance, the appropriate frame size, window type, and hop size must be selected. Often there is no clear choice. To address this we introduce the random-resolution STFT (RR), which randomly selects these parameters each time the loss is computed, ensuring the model is not biased by a fixed set of parameters.

Sum and difference loss — A loss function for stereo music was proposed in [6], which achieves left-right invariance by computing the sum and difference signals (Eq. 6) before applying the MR loss (Eq. 7), instead of directly operating on the left and right channels.

$$y_{\text{sum}} = y_{\text{left}} + y_{\text{right}} \quad y_{\text{diff}} = y_{\text{left}} - y_{\text{right}} \tag{6}$$

$$\ell_{S/D}(\hat{y}, y) = \ell_{\text{MR}}(\hat{y}_{\text{sum}}, y_{\text{sum}}) + \ell_{\text{MR}}(\hat{y}_{\text{diff}}, y_{\text{diff}}) \tag{7}$$

[1] `https://github.com/csteinmetz1/auraloss`

II. EVALUATION

To demonstrate the package, we train the same model each time using a different loss function. We employ a conditional temporal convolutional network (TCN) based on [7] for the task of modeling an analog dynamic range compressor [8]. The model is composed of 10 layers, each with kernel size 15, 32 channels, and exponentially increasing dilation factors for a receptive field of 324 ms at 44.1 kHz. We use Adam with a learning rate of $1 \cdot 10^{-3}$ and a batch size of 128, training each model for 20 epochs. We evaluate on the test set using all of the losses as error metrics as shown in Table 1.

Interestingly, we find that the lowest error for a given metric is not always achieved by optimizing that metric. It appears that training with a time domain loss leads to better performance on time domain metrics, with comparatively worse performance on frequency domain metrics, and vice versa. No formal conclusions can be made from this experiment, as differences in scaling of the losses during training may make comparisons challenging. We present this only as a demonstration of the package. Further work will examine these losses, and others, across more diverse audio generation tasks.

Model	**Test error**					
	L_1	ESR	Logcosh	STFT	MR	RR
L_1	**4.87e-3**	**0.0085**	**2.78e-5**	0.824	0.797	0.558
ESR	5.56e-3	0.0099	3.23e-5	0.806	0.779	0.549
Logcosh	5.30e-3	0.0093	3.03e-5	0.831	0.805	0.566
STFT	9.00e-3	0.0542	1.76e-4	0.451	0.432	0.339
MR	8.98e-3	0.0553	1.80e-4	**0.440**	**0.420**	**0.331**
RR	1.55e-2	0.2187	7.05e-3	0.525	0.504	0.392

Table 1: Test error across a model trained with different loss functions.

III. REFERENCES

[1] A. Wright, E.-P. Damskägg, V. Välimäki *et al.*, "Real-time black-box modelling with recurrent neural networks," in *DAFx*, 2019.

[2] A. Wright and V. Välimäki, "Perceptual loss function for neural modeling of audio systems," in *IEEE ICASSP*, 2020, pp. 251–255.

[3] P. Chen, G. Chen, and S. Zhang, "Log hyperbolic cosine loss improves variational auto-encoder," 2018.

[4] S. Ö. Arık, H. Jun, and G. Diamos, "Fast spectrogram inversion using multi-head convolutional neural networks," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 94–98, 2018.

[5] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *IEEE ICASSP*, 2020, pp. 6199–6203.

[6] C. J. Steinmetz *et al.*, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," *arXiv:2010.10291*, 2020.

[7] C. J. Steinmetz, "Learning to mix with neural audio effects in the waveform domain," Master's thesis, Universitat Pompeu Fabra, 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4091203

[8] S. Hawley, B. Colburn, and S. I. Mimilakis, "Profiling audio compressors with deep neural networks," in *AES*, 2019.