IMPROVING PEAK-PICKING USING MULTIPLE TIME-STEP LOSS FUNCTIONS

Carl Southall, Ryan Stables and Jason Hockman
DMT Lab, Birmingham City University, Birmingham, United Kingdom
{carl.southall, ryan.stables, jason.hockman}@bcu.ac.uk

ABSTRACT

The majority of state-of-the-art methods for music information retrieval (MIR) tasks now utilise deep learning methods reliant on minimisation of loss functions such as cross entropy. For tasks that include framewise binary classification (e.g., onset detection, music transcription) classes are derived from output activation functions by identifying points of local maxima, or peaks. However, the operating principles behind peak picking are different to that of the cross entropy loss function, which minimises the absolute difference between the output and target values for a single frame. To generate activation functions more suited to peak-picking, we propose two versions of a new loss function that incorporates information from multiple time-steps: 1) *multi-individual*, which uses multiple individual time-step cross entropies; and 2) *multi-difference*, which directly compares the difference between sequential time-step outputs. We evaluate the newly proposed loss functions alongside standard cross entropy in the popular MIR tasks of onset detection and automatic drum transcription. The results highlight the effectiveness of these loss functions in the improvement of overall system accuracies for both MIR tasks. Additionally, directly comparing the output from sequential time-steps in the multi-difference approach achieves the highest performance.

1. INTRODUCTION

At present, the state-of-the-art systems for many music information retrieval (MIR) tasks utilise deep learning models. Within the domain of dynamic time-series MIR tasks such as onset detection and music transcription, solutions are achieved through a binary classification of each time-step t. A binary classification output is typically limited to a range of [0,1] using a non-linear function (e.g., sigmoid, softmax). For classification purposes the output is subsequently rounded to either 0 or 1. However, in framewise binary classification tasks using this approach has proven to be less effective [7]. In the example presented in Figure 1, a framewise output activation function $\tilde{y}$ is shown in which the values ideally associated with a class label (i.e., value) of 1 do not exceed 0.5. While $\tilde{y}$ clearly shows the presence of an event as a *peak*, it would be identified as a false negative ($\tilde{y}^t < 0.5$).

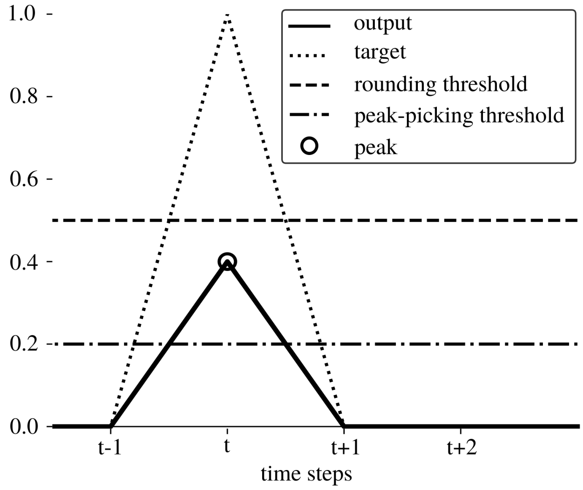


Figure 1. A true positive is missed using the rounding approach, but is successfully selected through peak picking (circled point). The solid line denotes the output, the dotted line the target, the dashed line the 0.5 rounding threshold and the dash-dotted line the peak-picking threshold.

1.1 Peak Picking

To overcome the problem posed in Figure 1, the majority of framewise binary classification systems utilise peak picking, which differentiates between classes by identifying local maxima. Multiple peak-picking approaches have been proposed in the literature [1, 4, 12, 16] and follow a general process as shown in Figure 1. Here, a point is selected as a peak if it is the maximum value within a local window and above a threshold τ. In [16] the threshold is determined by calculating the mean of a window, controlled using δ, a user determined constant λ and maximum and minimum values ($tmax$ and $tmin$).

$$\tau^t = mean(\tilde{y}^{t-\delta} : \tilde{y}^{t+\delta}) * \lambda \tag{1}$$

$$\tau^t = \begin{cases} tmax, & \tau > tmax \\ tmin, & \tau < tmin \end{cases} \tag{2}$$

Attribution: Carl Southall, Ryan Stables and Jason Hockman. "Improving Peak-picking Using Multiple Time-step Loss Functions", 19th International Society for Music Information Retrieval Conference, Paris, France, 2018.

An onset classification vector O is achieved by determining if each time-step of $\tilde{y}$ is the maximum value within the surrounding number of frames, set using Ω, and above the threshold τ:

$$O^t = \begin{cases} 1, & \tilde{y}^t == max(\tilde{y}^{t-\Omega} : \tilde{y}^{t+\Omega}) \quad \& \quad \tilde{y}^t > \tau^t \\ 0, & otherwise. \end{cases} \tag{3}$$

1.2 Loss Functions

The overall loss (often referred to as the *cost*) $\mathcal{L}$ represents the error of a system within a single value. It is calculated by comparing the difference between the desired ground truth y and the actual output $\tilde{y}$ [10]. Within audio based time-step classification tasks it is calculated by taking the mean of the individual time-step losses l^t:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} l^t. \tag{4}$$

$\mathcal{L}$ is a component of back propagation (and truncated back propagation) which is used to calculate the gradients $\mathcal{G}$ used in updating the trainable parameters of the model Θ with learning rate μ.

$$\Theta \leftarrow \Theta - \mu \cdot \mathcal{G} \tag{5}$$

Commonly used loss functions for calculating l^t include mean squared error (MS) (eq. 6) and cross entropy (CE) (eq. 7) [5].

$$l^t_{ms}\{y^t, \tilde{y}^t\} = (y^t - \tilde{y}^t)^2 \tag{6}$$

$$l^t_{ce}\{y^t, \tilde{y}^t\} = y^t \log(\tilde{y}^t) + (1 - y^t)\log(1 - \tilde{y}^t) \tag{7}$$

Both of these loss functions are suited to differentiating between binary classes using rounding as they aim to minimise the absolute difference between the targets y and the output $\tilde{y}$. In the majority of MIR tasks CE is more suited than MS due to its greater penalization of large errors [14, 16, 22].

1.3 Motivation

In the peak-picking process, multiple frames are utilized in both the calculation of a threshold as well as the peak selection. However, in the MS and CE calculations only the current time-step t is used in measuring the difference between the target y and output $\tilde{y}$. In order for the loss to reflect peak salience (i.e., the clarity of the local maxima) and to ensure that the output activation function is suitable for peak-picking, then multiple time-steps should be included within the loss function calculation. To this end, we propose two versions of a new loss function which not only measures the absolute difference between y and $\tilde{y}$, but also allows for peak salience to be maintained. We then evaluate the worth of these functions within the tasks of onset detection and automatic drum transcription (ADT).

The remainder of this paper is structured as follows: Section 2 presents the proposed loss functions and Section 3 gives an overview of the evaluation. The results and discussion are presented in Section 4 and the conclusions and future work are presented in Section 5.

2. METHOD

For a loss function to represent an understanding of peak salience, it must include at least three points: $\tilde{y}^{t-1} : \tilde{y}^{t+1}$. We propose combining CE and a peak salience measure into a single loss function termed peakiness cross entropy (PCE):

$$l^t_{pce} = \frac{1}{2}\big(\gamma l^t_{ce}\{y^t, \tilde{y}^t\} + (1-\gamma)(l^t_p + l^t_f)\big), \tag{8}$$

where the first part of the equation is the standard cross entropy (CE) of the current time-step t. The second part of the function is a peak salience measure that consists of two variables: l_p, which focuses on the previous time-step and l_f, which focuses on the future $(t+1)$ time step. γ is used to control the weighting between standard CE and the peakiness measure. We propose two methods for achieving l_p and l_f: a combination of multiple individual time-step calculations and a direct comparison of the differences between multiple time-steps.

2.1 Multi-individual

The multi-individual (MI) method calculates l_p and l_f as individual time step cross entropies of previous and future time-steps:

$$l^t_p = l^t_{ce}\{y^{t-1}, \tilde{y}^{t-1}\} \tag{9}$$

$$l^t_f = l^t_{ce}\{y^{t+1}, \tilde{y}^{t+1}\}. \tag{10}$$

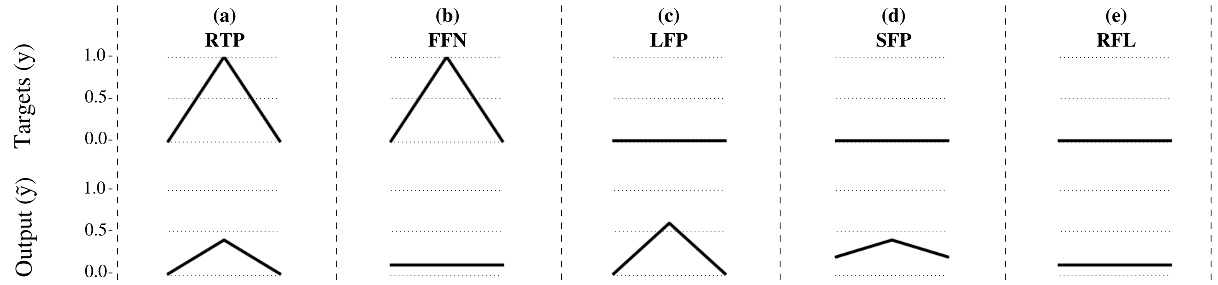
This ensures that updates to $\tilde{y}^t$ do not cause greater negative updates to $\tilde{y}^{t-1}$ and $\tilde{y}^{t+1}$.

2.2 Multi-difference

Although MI utilizes multiple time-steps it does not compare absolute differences between sequential time-steps. To achieve this, we propose an additional calculation of l_p and l_f, termed multi-difference (MD), which measures the absolute differences between sequential time-steps of the target y and the output $\tilde{y}$. The first version of MD (MMD), utilizes MS. The second version (WMD) utilizes an updated version of the CE equation, termed weighted cross entropy (WCE):

$$l^t_{wce}\{y^t, \tilde{y}^t\} = (1-\phi)y^t \log(\tilde{y}^t) + \phi(1-y^t)\log(1-\tilde{y}^t), \tag{11}$$

which allows the strength of each half of the equation to be controlled using the weighting parameter ϕ. The first half of the WCE equation (henceforth referred to as WCE-FN) aims to reduce false negatives by producing a loss value proportional to the difference between sequential time-steps of y^t and $\tilde{y}^t$. The second half of the WCE equation (hereafter termed WCE-FP) aims to suppress false positives



Loss Funcs.	(a) RTP	(b) FFN	(c) LFP	(d) SFP	(e) RFL
CE	0.92	2.30	0.92	0.51	0.11
MI	0.46	1.26	0.46	0.48	0.16
MMD	0.01	0.15	0.10	0.22	0.05
WMD	0.92	19.17	0.92	0.37	0.05
WMD-FN	1.37	37.19	0.46	0.26	0.05
WMD-FP	0.47	1.15	1.37	0.48	0.05

Figure 2. Example activation function scenarios with corresponding loss values output from each loss function. From left to right: a raised true positive (RTP), a flat line false negative (FFN), a large false positive (LFP), a small raised false positive (SFP) and a raised flat line (RFL).

as it outputs a larger value if there is a large undesirable difference between sequential frames. l_p^t and l_f^t in MMD and WMD are calculated respectively using:

$$l_p^t = \begin{cases} l_{wce}^t\{|y^t - y^{t-1}|, |\tilde{y}^t - \tilde{y}^{t-1}|\}, & WMD, \\ l_{ms}^t\{|y^t - y^{t-1}|, |\tilde{y}^t - \tilde{y}^{t-1}|\}, & MMD, \end{cases} \tag{12}$$

$$l_f^t = \begin{cases} l_{wce}^t\{|y^t - y^{t+1}|, |\tilde{y}^t - \tilde{y}^{t+1}|\}, & WMD, \\ l_{ms}^t\{|y^t - y^{t+1}|, |\tilde{y}^t - \tilde{y}^{t+1}|\}, & MMD. \end{cases} \tag{13}$$

Truncated back propagation is used to calculate the gradients for all loss functions. The presented implementation utilises the automatic differentiation functions built into the Tensorflow [1] library for this purpose.

2.3 Example Loss Function Scenarios

Figure 2 presents five example activation function scenarios. The loss values achieved by CE, MI, MMD, WMD and the two separated halves of WMD: WMD-FN ($\phi = 0$) and WMD-FP ($\phi = 1$), are presented with $\gamma = 0.5$. The targets are presented at the top and the output activation function on the bottom. It is worth noting that all of the loss functions that utilize CE can be directly compared but MMD is relative to itself (i.e., the MMD values might seem small relative to the other loss values but not relative to other values of MMD). If all frames of the output are correct then all of the loss functions output zero.

(a) ***Reduced true positive***: The first example shows a reduced true positive where the surrounding frames are correct. In this case CE and WMD output the largest values as this peak could fall below the peak-picking threshold.

(b) ***Flat line false negative***: The second example shows a false negative where the output is a flat line. In this case high relative error values are given by all of the loss functions, however larger error values are given by MMD and especially the FN suppression half of WMD. This example would generally not be selected during peak-picking.

(c) ***Large false positive***: The third example shows a false positive where the surrounding frames are correct. In this case high values are given by CE, MMD and the false positive suppression part of WMD, as this would be an incorrectly selected peak.

(d) ***Small raised false positive***: The fourth example again shows a false positive, similar to the previous example, but the surrounding frames are raised resulting in a less salient false positive. In this case lower values are given by MI and WMD-FP, than CE, as this peak is not as salient as the one in example three (i.e., large false positive).

(e) ***Raised flat line***: The final example presents a raised flat line. In this case the MMD and WMD loss functions penalize less than CE and MI. While the absolute values are slightly wrong, the difference between the sequential frames is correct, resulting in no peaks being correctly chosen.

3. EVALUATION

To identify whether the new loss functions improve performance, we compare the newly proposed loss functions against standard cross entropy (CE) in two common MIR tasks: onset detection (OD) and automatic drum transcription (ADT). To ensure performance trends are consistent with different systems, we implement four neural network

[1] https://www.tensorflow.org

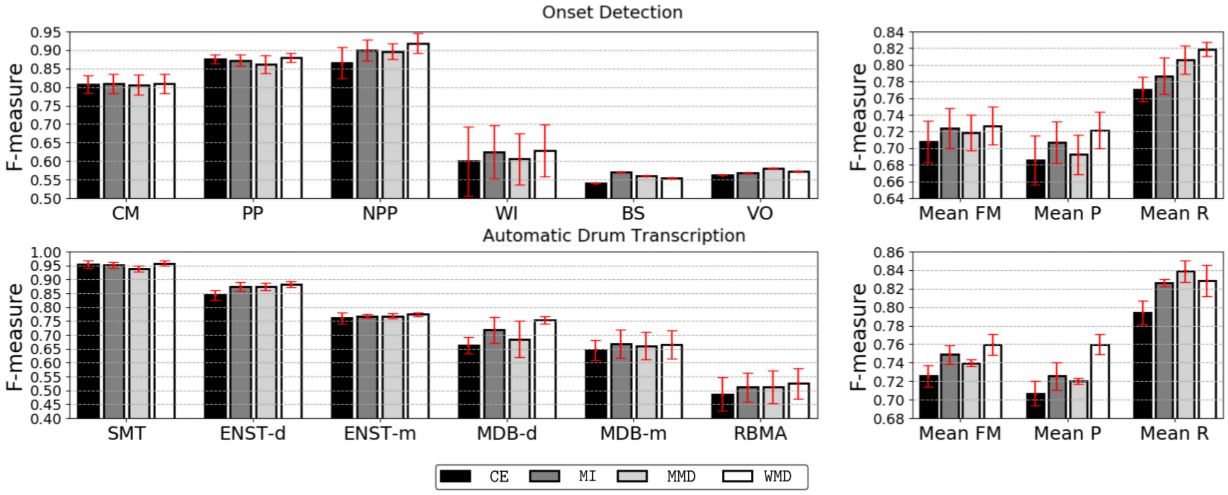


Figure 3. Subset mean system F-measure results for the four implemented cost functions for onset detection and automatic drum transcription. The individual subset F-measure results are on the left and the mean subset F-measure, precision and recall results are on the right. The red error plots display the standard deviations across the folds.

based models which have achieved state-of-the-art results for both of the tasks in recent years. Standard F-measure, derived from precision and recall, is used as the evaluation metric with onset candidates being accepted if they fall within 30ms of the ground truth annotations (i.e., window of acceptance). If onset candidates fall within 30ms of each other, they are combined into a single onset at the middle location (i.e., window of combination).

3.1 Onset Detection (OD)

For the OD evaluation, we utilize the same datasets and subset splits as used in [3], consisting of: complex mixtures (CM), pitched percussion (PP), non-pitched percussion (NPP), wind instruments (WI), bowed strings (BS) and vocals (VO). As OD is a binary classification task, all systems are implemented with a two neuron softmax output layer, one neuron corresponds to an onset and the other neuron corresponds to the absence of an onset.

3.2 Automatic Drum Transcription (ADT)

For the ADT evaluation, we utilize four ADT datasets: IDMT-SMT-Drums [6], ENST-Drums *minus one* subset [8], MDB-Drums [18] and RBMA-2013 [21]. To observe trends between contexts, the datasets are divided into the three groups proposed in [23]: 1) drum only (DTD) consisting of IDMT-SMT-Drums, 2) drums in the presence of extra percussion (DTP) consisting of the drum-only versions ENST-d and MDB-d and 3) drums in the presence of extra percussion and melodic instruments (DTM), which consist of the polyphonic versions ENST-m, MDB-m and RBMA-2013. ENST-m is created by combining the ENST drum tracks and the accompaniment files using ratios of $\frac{2}{3}$ and $\frac{1}{3}$ respectively, as done in [6, 9, 15, 20, 24]. A three-neuron sigmoid output layer is used for all implemented ADT systems, with the neurons corresponding to the three observed drum instruments (i.e., KD, SD and HH).

3.3 Systems

Four different neural network based systems are implemented. All systems consist of the same overlying structure: First, input features are fed into a pre-trained neural network model. Peak-picking is then performed to determine the locations of the onset candidates using the algorithm from [16] (eq.1:3).

3.3.1 Input Features

For both tasks we use the same framewise logarithmic spectral input features x generated using the madmom Python library [2]. The input audio (16-bit .wav file sampled at 44.1 kHz) is segmented into T frames using a Hanning window of m samples (m = 2048) with a $\frac{m}{2}$ hopsize. A logarithmic frequency representation of each of the frames is created using a similar process to [22]. The magnitudes of a discrete Fourier are transformed into a logarithmic scale (20Hz–20kHz) using twelve triangular filters per octave. This results in a 84 x T logarithmic spectrogram.

3.3.2 lstmpB

The `lstmpB` system is based on the system presented in [23] and the baseline system used in [16]. It consists of two 50-neuron hidden layers containing long short-term memory cells with peephole connections. The input features are processed in a framewise manner.

3.3.3 lstmpSA3B

The `lstmpSA3B` system is based on the SA3 system proposed in [16]. It is the same as the `lstmpB` system other than it contains a soft attention mechanism in the output layer. As in the original implementation the attention number a controls the number of attention connections, and is set to three.

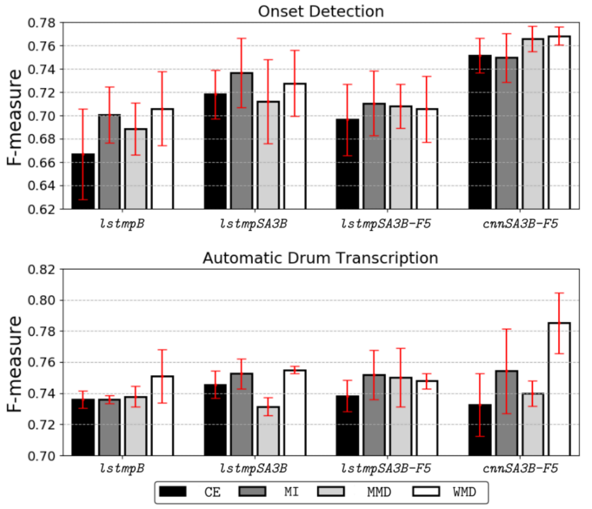


Figure 4. Individual system mean subset F-measure results for the proposed cost functions in OD and ADT tasks. Red bars denote standard deviations across folds.

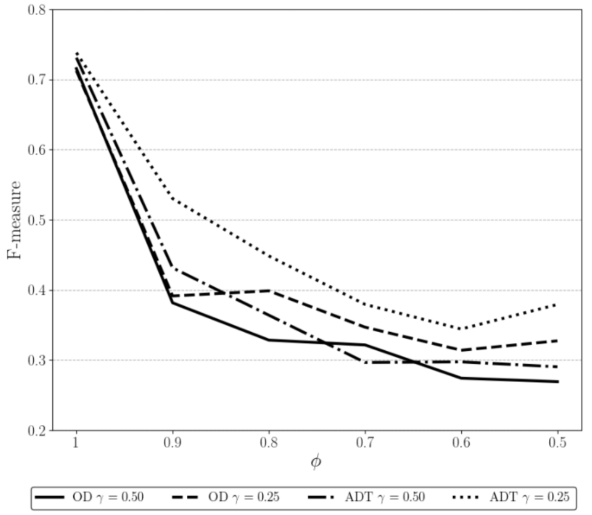


Figure 5. Mean system and mean subset results for different values of WMD parameters (γ and ϕ) in onset detection (OD) and automatic drum transcription (ADT) evaluations.

3.3.4 lstmpSA3B-F5

The *lstmpSA3B-F5* system is identical to the *lstmpSA3B* system with a larger number of input features used. A total of 11 frames (5 either side of the current frame (x^{t-5} : x^{t+5})) are used for each time-step.

3.3.5 cnnSA3B-F5

The *cnnSA3B-F5* [17] combines the convolutional recurrent neural network proposed in [22] and the soft attention mechanism proposed in [16]. It contains two convolutional layers consisting of 3x3 filters, 3x3 max pooling, dropouts [19] and batch normalization [11], with the first layer consisting of 32 channels and the second 64 channels. It contains the same soft attention mechanism output layer and the same input feature size as *lstmpSA3B-F5*.

3.3.6 Training

All systems are trained using mini-batch gradient descent with the Adam optimizer [13]. An initial learning rate of 0.003 is used and three-fold cross validation is performed. Each mini-batch consists of 10 randomly chosen, 100 time-step segments and the data is divided by track into 70% training, 15% validation and 15% testing sets. The training data is used to optimize the systems and the validation data is used to prevent overfitting and to optimize the peak-picking parameters. For datasets containing subsets (i.e., IDMT-SMT Drums and ENST Drums) the splits are performed evenly across the subsets.

4. RESULTS AND DISCUSSION

4.1 Subset Performance

Figure 3 presents the subset results for all cost functions in both evaluations. The red error bars represent standard deviation across folds. The OD results are derived from the mean of the systems and the ADT results are derived from the mean of the systems and the mean of the observed drum instruments (i.e., KD, SD and HH). The left part of the figure presents the individual subset F-measures and the right part of the figure presents the mean subset F-measure, precision and recall. In both MIR tasks, all three of the newly proposed cost functions achieved a higher mean subset F-measure than standard CE, with WMD performing the best in both. Within the ADT evaluation a higher performance is achieved for all three observed drum instruments. A slightly larger increase in performance was witnessed in the ADT task and both versions of the MD cost function achieve higher mean subset F-measures than MI. This highlights that measuring the absolute differences between sequential frames does improve performance. The mean subset precision and recall results highlight that in all cases the newly proposed cost functions achieve higher precision and recall scores than standard CE. In the OD evaluation the highest increase in performance between WMD and CE is in the NPP subset. In the ADT evaluation the largest increase is seen within the DTP subsets (ENST-d and MDB-d). For all subsets in both evaluations the highest F-measure is achieved by one of the three newly proposed cost functions and the error bars show that this improvement occurs in all of the folds. Results from t-tests highlight that the WMD systems improvement over CE within the BS and mean recall OD categories and MDB-d, mean F-measure and precision ADT categories are significant ($\rho < 0.05$).

4.2 Individual System Performance

Figure 4 presents the individual system, mean subset F-measure results for both MIR evaluations. In all cases the highest F-measure is achieved by one of the newly proposed cost functions, with the WMD cost function achieving the highest F-measure in five of the eight cases. This reinforces that using multiple framed cost functions does improve performance and that this increase is not just associated with one system. The highest F-measure and the largest increase relative to CE is achieved by the *cnnSA3B-F5* system using the WMD cost function.

4.3 WMD Parameters

Figure 5 presents the mean system, mean subset F-measure results for different parameter settings of the WMD cost function. Plots of six ϕ values for $\gamma = 0.25$ and $\gamma = 0.5$ are presented for both evaluations. For any ϕ values less than one, there is a dramatic decrease in performance which suggests that the false negative suppression half of the WCE function has a negative effect on performance. This is possibly due to the extremely high value given to flat parts of the activation function (see Figure 2), causing these parts of the activation function to become noisy. This suggests that the improvement is due to the false positive suppression half of the WMD system. As this alone achieves higher F-measures than the other proposed cost functions, then it also suggests that their improvement is also due to the suppression of false positives. However, the false negative suppression in those cost functions do not cause repercussions. As γ (i.e., weighting of peak salience measure) increases (0.5 = even weighting with standard CE) then the performance decreases. This trend continues with all values below 0.5 and with the other two proposed loss functions. The highest F-measures were achieved with γ=0.25 (Standard CE is weighted twice as much as the peak salience measure) for all three proposed loss functions. To categorically identify ideal parameter settings for a particular scenario, a grid search would be required. However, the results suggest that $\gamma = 0.25$ and $\phi = 1$ would always be optimal.

4.4 Understanding the Improvement

After visual comparison of the output activation functions a common situation in which the newly proposed loss functions achieve higher performance was observed. Figure 6 presents an example of this situation, with the top diagram showing the output activation function using CE and the bottom diagram showing the activation function when using the highest performing version of WMD. In the CE diagram, there are two spikes to the right that are wrongly detected as peaks but in the WMD version these peaks are less salient, resulting in no false positives. The consequence of this is that the actual true positive within in the WMD version has a lower amplitude than the one in the CE version. However, this has no effect on performance as the true positive is still a clear peak and correctly chosen within both CE and WMD versions. We believe this situation occurs because within CE a higher error is given to the true positive than the combination of the two smaller false positive errors. This causes the true positive to be closer to the target y but consequentially causes the false positive spikes. Within the WMD version, the false positive suppression assigns a greater loss value to the two false positive spikes than the reduced true positive, ensuring that the spikes are not selected by the peak-picking algorithm. This reduction of noise in the activation function results in less false positives but also enables the peak-picking threshold to be lower, enabling more true positives to be selected. This effect could likely explain both the increase in recall and precision.

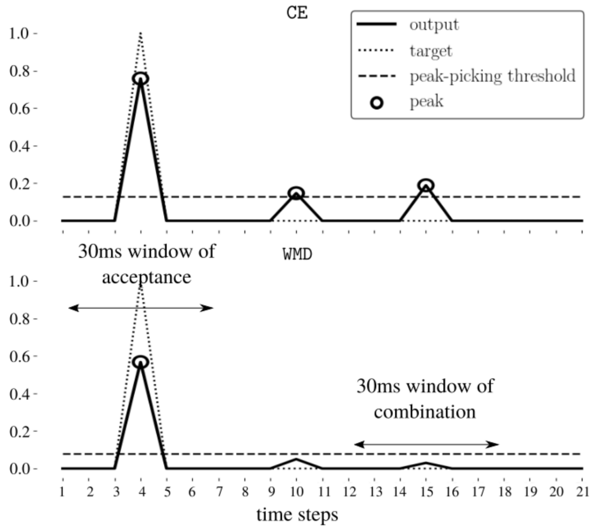


Figure 6. Example of WMD loss function reducing the number of false positives by suppressing false spikes. CE output activation function (top) and WMD output activation function (bottom) with output $\tilde{y}$ (solid lines), target y (dotted lines) and the peak-picking threshold (dashed lines). Circles denote selected peaks and arrowed lines show windows of acceptance and combination.

5. CONCLUSIONS AND FUTURE WORK

We have developed three new loss functions in an attempt to generate activation functions more suited to peak-picking. The new loss functions utilise information from multiple time-steps which allow them to measure both the absolute values and to maintain peak salience by comparing sequential time-steps. We evaluated the newly proposed loss functions against standard CE using four neural network-based systems in the MIR tasks of onset detection and ADT. The results highlight that all three of the newly proposed cost functions do improve performance, with the WMD loss function achieving the highest accuracy. This work focuses on the inclusion of a single frame on either side of the current time-step. Future work could explore the potential benefit of using a greater number of frames and a version of the WMD equation in which the false negative suppression component does not negatively influence the outcome. Additionally, to make the system end-to-end, the evaluation methodology (i.e., F-measure and tolerance windows) could also be incorporated within the loss functions. Open source implementations of the new loss functions are available online. [2]

[2] https://github.com/CarlSouthall/PP_loss_functions

6. REFERENCES

[1] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 13(5):1–13, 2005.

[2] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. madmom: A new Python audio and music signal processing library. In *Proceedings of the ACM International Conference on Multimedia*, pages 1174–1178, Amsterdam, The Netherlands, 2016.

[3] Sebastian Böck, Florian Krebs, and Markus Schedl. Evaluating the online capabilities of onset detection methods. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, 2012.

[4] Sebastian Böck, Jan Schlüter, and Gerhard Widmer. Enhanced peak picking for onset detection with recurrent neural networks. In *Proceedings of the 6th International Workshop on Machine Learning and Music (MML)*, pages 15–18, Prague, Czech Republic, 2013.

[5] Pieter-Tjerk de Boer, Dirk Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 1 2005.

[6] Christian Dittmar and Daniel Gärtner. Real-time transcription and separation of drum recordings based on NMF decomposition. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 187–194, Erlangen, Germany, 2014.

[7] Florian Eyben, Sebastian Böck, Björn Schuller, and Alex Graves. Universal onset detection with bidirectional long-short term memory neural networks. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 589–594, Utrecht, The Netherlands, 2010.

[8] Olivier Gillet and Gaël Richard. Enst-drums: an extensive audio-visual database for drum signals processing. In *Proceedings of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, pages 156–159, Victoria, Canada, 2006.

[9] Olivier Gillet and Gaël Richard. Transcription and separation of drum signals from polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):529–540, 2008.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[12] Ismo Kauppinen. Methods for detecting impulsive noise in speech and audio signals,. In *Proceedings of the 14th International Conference on Digital Signal Processing (DSP2002)*, pages 967–970, Santorini, Greece, 2002.

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

[14] Jan Schluter and Sebastian Bock. Improved musical onset detection with convolutional neural networks. In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6979–6983, 2014.

[15] Carl Southall, Ryan Stables, and Jason Hockman. Automatic drum transcription using bi-directional recurrent neural networks. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 591–597, New York City, United States, 2016.

[16] Carl Southall, Ryan Stables, and Jason Hockman. Automatic drum transcription for polyphonic recordings using soft attention mechanisms and convolutional neural networks. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 606–612, Suzhou, China, 2017.

[17] Carl Southall, Ryan Stables, and Jason Hockman. Player vs transcriber: A game approach to data manipulation for automatic drum transcription. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018.

[18] Carl Southall, Chih-Wei Wu, Alexander Lerch, and Jason Hockman. MDB drums an annotated subset of medleyDB for automatic drum transcription. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017.

[19] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[20] Richard Vogl, Matthias Dorfer, and Peter Knees. Recurrent neural networks for drum transcription. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 730–736, New York City, United States, 2016.

[21] Richard Vogl, Matthias Dorfer, and Peter Knees. Drum transcription from polyphonic music with recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 201–205, New Orleans, United States, 2017.

[22] Richard Vogl, Matthias Dorfer, Gerhard Widmer, and Peter Knees. Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 150–157, Suzhou, China, 2017.

[23] Chih-Wei Wu, Christian Dittmar, Carl Southall, Richard Vogl, Gerhard Widmer, Jason Hockman, Meinard Müller, and Alexander Lerch. A Review of Automatic Drum Transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9):1457–1483, 2018.

[24] Chih-Wei Wu and Alexander Lerch. Drum transcription using partially fixed non-negative matrix factorization with template adaptation. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 257–263, Malaga, Spain, 2015.