INVESTIGATING LABEL NOISE SENSITIVITY OF CONVOLUTIONAL NEURAL NETWORKS FOR FINE GRAINED AUDIO SIGNAL LABELLING

Rainer Kelz *Gerhard Widmer*

Johannes Kepler University Linz
Department of Computational Perception
Altenberger Str. 69, 4040 Linz, Austria

ABSTRACT

We measure the effect of small amounts of systematic and random label noise caused by slightly misaligned ground truth labels in a fine grained audio signal labeling task. The task we choose to demonstrate these effects on is also known as *framewise polyphonic transcription* or *note quantized multi-f0 estimation*, and transforms a monaural audio signal into a sequence of note indicator labels. It will be shown that even slight misalignments have clearly apparent effects, demonstrating a great sensitivity of convolutional neural networks to label noise. The implications are clear: when using convolutional neural networks for fine grained audio signal labeling tasks, great care has to be taken to ensure that the annotations have precise timing, and are free from systematic or random error as much as possible - even small misalignments will have a noticeable impact.

Index Terms— convolutional neural networks, multi-label classification, framewise polyphonic transcription

1. INTRODUCTION

Recent empirical work on quantifying the generalization capabilities of deep neural networks [1, 2] questions the usefulness of traditional learning theory applied to neural networks, and among other things shows that image classification performance gradually degrades in the presence of mislabeled images.

We investigate how severe these effects are for time series labeling when the ground truth labels are only slightly misaligned, which is a common occurrence when dealing with manually annotated time series data. The assumption being that the main difference between label noise in time series and label noise for images caused by annotators is the similarity of examples in input space. This comparison is justified by the fact that the way a sequence labeling task with convolutional neural networks is usually set up, corresponds exactly to repeated image classification on similar images obtained by shifting a reading window across the short time Fourier transformed audio signal. However, because the distribution of examples in the image domain is different from examples obtained in the audio domain it is not immediately clear to which extent label noise is a problem.

We posit that it is highly unlikely that two similar images will be assigned different labels by the same annotator. It is more likely that an imprecision either in hand movements steering a pointing device such as the mouse, or the annotation software used itself, will lead to very similar examples in time being assigned different labels. A sketch of this intuitive notion can be seen in figure 1, where we can observe the sequential transformations of the audio signal input together with its annotation.

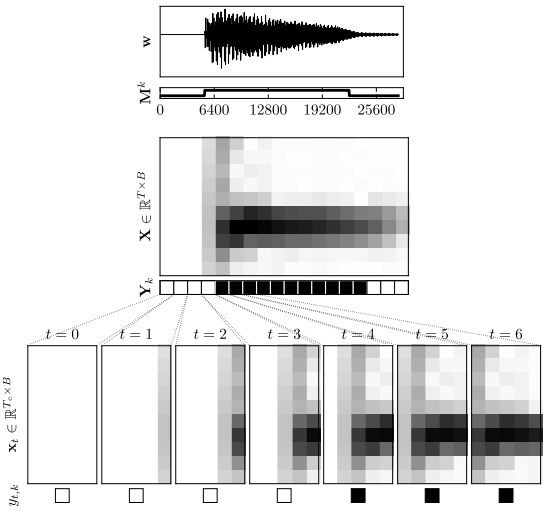


Fig. 1: The intuitive reason why imprecision in audio signal annotation may yield highly similar, yet differently labeled examples. At the last stage of the signal processing chain we see pairs of input and corresponding indicator $(\mathbf{x}_t, y_{t,k})$ for label k. Note the very similar frames $\mathbf{x}_3$ and $\mathbf{x}_4$ and their different labels.

The data consists of pairs $\mathbf{w} \in \mathbb{R}^{T^{hi}}$ denoting the audio signal and $\mathbf{M} \in \{0,1\}^{T^{hi} \times K}$ denoting the annotation, where T^{hi} is the number of audio samples, K is the number of labels, and k is the index of an arbitrary label. A common pre-

processing step is the short-time Fourier transform (STFT) of $\mathbf{w}$ and subsequent application of a filter bank to obtain $\mathbf{X} \in \mathbb{R}^{T^{lo} \times B}$. B is the frequency resolution, dependent on the choice of filter bank applied after the Fourier transform. The usually much lower frame rate of the STFT is dependent on the hop size, resulting in $T^{lo} \ll T^{hi}$. In a similar fashion, the high resolution annotation $\mathbf{M}$ is transformed into $\mathbf{Y} \in \mathbb{R}^{T^{lo} \times K}$ to match up with the filtered STFT.

The final inputs to the convolutional neural network are pairs $\mathbf{x}_t \in \mathbb{R}^{T_c^{lo} \times B}$ of excerpts of length T_c^{lo} from $\mathbf{X}$ at time t and labels $y_{t,k} \in \{0, 1\}$ as a target for each label indicator output. Looking at these steps in detail makes it apparent that slight misalignments in the annotation, be they random or systematic, lead to collections of pairs $\{(\mathbf{x}_a, y_{a,k}), (\mathbf{x}_b, y_{b,k}), \dots\}$ where most distance measures in input space $d(\mathbf{x}_a, \mathbf{x}_b)$ are small but the targets for these examples differ, as $y_{a,k} \neq y_{b,k}$.

The sequence labeling task we chose to investigate the impact of misaligned annotations on, is also called *framewise polyphonic transcription* or *note quantized multi-f0 estimation* in the music information retrieval community. We claim that the effects measured here also extend to beyond the framewise scenario because the misalignment problems we consider only affect the start and end positions of a label, and hence also extend to systems that try and predict labels in interval form.

2. MODELS

The convolutional neural networks we use for time series labeling are parametrized functions $g_\theta : \mathbb{R}^{T_c \times B} \to \{0, 1\}^K$, mapping excerpts from a filtered STFT of length T_c in time and width B in frequency to a vector of length K, whose components indicate the presence or absence of a label. After the application of a logarithmic filter bank to the STFT, the number of bins comes down to $B = 229$, as described in [3]. For framewise transcription of pianos, $K = 88$ denotes the tonal range of the instrument, and a label indicator having a value of 1 means that a note is sounding in the excerpt presented as the input. The misalignment effects are measured at two different STFT frame rates, $31.25\,[\text{fps}]$ and $100\,[\text{fps}]$, the lower frame rate also being used in [4, 3]. Keeping the temporal context approximately the same for the two frame rates necessitated the use of two different architectures, with the one for the higher frame rate being deeper and wider, yet only slightly increasing parameter count. [1]

For the training procedure, we adhere closely to the description in [3], which uses mini-batch stochastic gradient descent with Nesterov momentum and a step-wise learning rate schedule, but reduced mini-batch size. A rather drastic reduction of the number of examples in a batch from 128 examples as advocated in [3], to 8 examples in the present work, is motivated by findings in [5], which state that noisier gradient estimates are helpful in finding flatter minima, thus potentially improving generalization. We notice a small improvement in prediction performance together with a convenient reduction in training time.

[1] Source code to replicate all results can be found at `https://github.com/rainerkelz/ICASSP18`

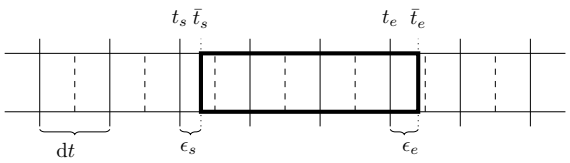
3. DATASET

We chose the MAPS dataset [6] as our experimental testbed due to the availability of a very precise ground truth, free from any human annotator disagreement. Note that for the purpose of demonstrating non-negligable effect sizes of misalignments, any annotation could suffice in principle, as long as it is unambiguous. The data consists of a collection of MIDI files, and corresponding audio renderings. Multiple sample banks were used to render the audio files. To further increase acoustic variability, several MIDI files were played back on a computer controlled Disklavier and recorded in close and ambient microphone conditions. The MIDI files in the MAPS dataset have a sufficiently high temporal resolution that enables us to neglect any quantization error stemming from the conversion of MIDI ticks to seconds and treat the start and end times of note labels effectively as if they were originating from a continuous space. There are two train-test protocols defined in [4], with different amounts of instrument overlap. In *Configuration-I*, instruments in training, validation and test sets overlap, whereas in *Configuration-II* only training and validation sets contain overlapping instruments. The test set for *Configuration-II* solely consists of pieces rendered with the Disklavier. For both configurations, the $31.25\,[\text{fps}]$ models are trained and evaluated on four different splits of the training data. The models with the higher frame rate input at $100\,[\text{fps}]$ are trained and evaluated on one fold only for both configurations due to the high computational cost.

4. EXPERIMENTAL SETUP

For all models trained, all non-architectural hyper parameters are fixed, the only varied quantities are the choice of frame rate and the choice of labeling function, the exact notion of which we will now define.

Throughout this paper we will use the term *labeling function* to mean the complete conversion process from high resolution annotations to assigning concrete labels $y_{t,k} \in \{0, 1\}$ to the examples $\mathbf{x}_t \in \mathbb{R}^{T_c \times B}$ shown to the network. This includes the annotator as well, be it a mechanistic generator, as in the case of extracting labels from MIDI files, or a human providing manual annotation. The main focus for each function lies on the conversion from high resolution annotations to lower resolution annotations. Figure 2a shows a schematic depiction of the conversion of a label from an interval defined



(a) schematic illustration of quantities involved in the definition of labeling functions for fine grained sequence labeling tasks

$f_\cdot(\bar{t}_s, \bar{t}_e)$	t_s	t_e
f_a	$\lfloor \bar{t}_s / \, dt \rceil$	$\lfloor \bar{t}_e / \, dt \rceil$
f_b	$\lceil \bar{t}_s / \, dt \rceil$	$\lceil \bar{t}_e / \, dt \rceil$
f_c	$\lfloor \bar{t}_s / \, dt \rfloor$	$\lfloor \bar{t}_e / \, dt \rfloor$
f_d	$\lfloor \bar{t}_s / \, dt \rfloor$	$\lfloor \bar{t}_e / \, dt \rceil + \lfloor (\bar{t}_e - \bar{t}_s) / \, dt \rfloor$
f_e	$f_a + R_j$	$f_a + R_j$
f_f	$f_a + R_s$	$f_a + R_e$

(b) The different labeling functions used, leading to different kinds of quantization error. Symbols $\lfloor \cdot \rfloor, \lfloor \cdot \rceil, \lceil \cdot \rceil$ denote the functions $\mathrm{floor}(\cdot), \mathrm{round}(\cdot), \mathrm{ceil}(\cdot)$ respectively, and $R_{j,s,e} \in \{-1, 0, 1\}$ are discrete, uniformly distributed random variables. f_a is used as the reference labeling function throughout this article.

Fig. 2: Quantization schemes and definitions of labeling functions.

with high time resolution into a sequence of much fewer labels at a lower time resolution. Symbols $\bar{t}_s, \bar{t}_e$ denote start and end times in high resolution, expressed in seconds (we pretend these are continuous, due to their high time resolution). Symbols t_s, t_e are their counterparts in low resolution, expressed as discrete frame indices, with ϵ_s, ϵ_e denoting the errors incurred by rounding after conversion. The symbol dt denotes the length of one lower resolution frame in seconds and is used as the conversion factor.

The exact definitions of the different labeling functions used to transform the high resolution annotations obtained from MIDI files into framewise labels can be found in figure 2b. They can all be viewed as functions of the form $f : \mathbb{R} \times \mathbb{R} \to \mathbb{N} \times \mathbb{N}$, mapping pairs of continuous times to pairs of discrete indices.

The differences between labeling functions lie in the choice of how to quantize and when. Functions $f_{\{a,b,c,d\}}$ deal with systematic misalignment caused by systematic quantization errors, and functions $f_{\{e,f\}}$ randomly modify the result of f_a, by either shifting both start and end indices jointly by a random variable $R_j \in \{-1, 0, 1\}$, or shifting the two indices separately by two independent random variables $R_s, R_e \in \{-1, 0, 1\}$. For all functions involving random variables, their realizations are drawn from a discrete uniform distribution at the time of conversion. This means that for each pair of start and end times and a particular experimental run, a potential shift can happen only once.

We treat the labeling function f_a, which simply rounds to the nearest integer, as the reference. All evaluations are done on the first 30[s] of all pieces in the respective test set for a fold, and against a ground truth obtained at a frame rate of 100 [fps] with labeling function f_a. This means the predictions of the models running at a lower frame rate of 31.25 [fps] need to be upsampled again for evaluation. To measure prediction performance, we use the definitions for precision $\mathcal{P}$, recall $\mathcal{R}$ and f-measure $\mathcal{F}$ as described in [7].

$$\mathcal{P} = \frac{\sum_{t=1}^{T} TP[t]}{\sum_{t=1}^{T} TP[t] + FP[t]} \tag{1}$$

$$\mathcal{R} = \frac{\sum_{t=1}^{T} TP[t]}{\sum_{t=1}^{T} TP[t] + FN[t]} \tag{2}$$

$$\mathcal{F} = \frac{2 \cdot \mathcal{P} \cdot \mathcal{R}}{\mathcal{P} + \mathcal{R}} \tag{3}$$

5. RESULTS

The effect of small misalignments on label annotation can be observed in figures 3 and 4, which show clear evidence of the effect of using different labeling functions at a frame rate of 31.25 [fps], across multiple folds for Configuration I with similar data in train and test sets. Along the horizontal axis we see the labeling functions, and the vertical axis shows f-measure.

For each labeling function we see the performance on individual folds ($\times, \circ, \square, \diamond$) and the textually annotated mean ($-$) of all folds. We immediately notice that a labeling function with either a small systematic ($f_{\{a,b,c,d\}}$) or random ($f_{\{e,f\}}$) error has a non-negligible effect on the f-measure.

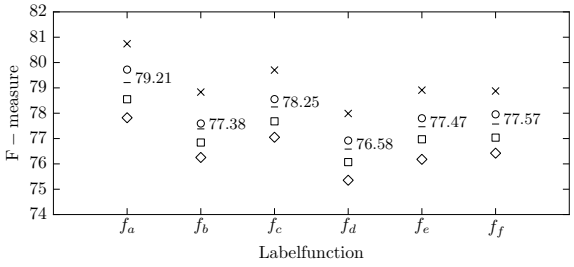


Fig. 3: Configuration I: F-measure on the validation set at a frame rate of 31.25 [fps] across different labeling functions for multiple folds.

The impact on the performance on the test set is less severe, as can be observed in figure 4, but still on the order of 1 percentage point. Incidentally, the mean result for the reference labeling function improves slightly on the state of the art for this task as reported in [3].

Interestingly, when inspecting the lower frame rate results for Configuration II which has much more dissimilar train and

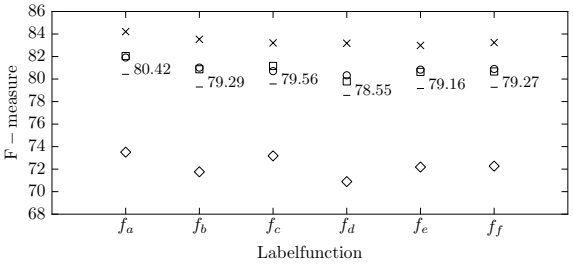


Fig. 4: Configuration I: F-measure on the test set at a frame rate of 31.25 [fps] across different labeling functions for multiple folds.

test sets in terms of acoustic conditions, we can still observe a similar pattern. Even small misalignments lead to non negligible differences in performance, observable on the test set results in figure 5. The obtained results all improve upon the state of the art as reported in [3], regardless of labeling function which we attribute to some extent to the much smaller batch size and the usage of a labeling function introducing systematic error in [3].

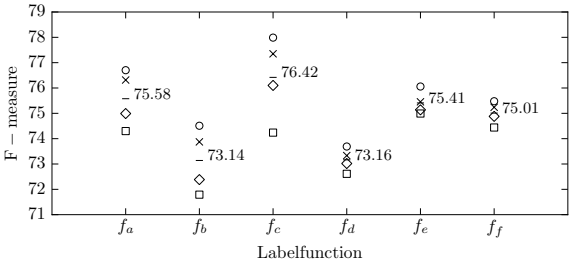


Fig. 5: Configuration II: F-measure on the test set at a frame rate of 31.25 [fps] across different labeling functions for multiple folds.

Finally, focusing our attention on the results for the test set at a higher frame rate of 100 [fps] in figure 6, a pattern of performance differences with respect to the reference labeling function f_a is still noticeable. The severity is much smaller however, which is to be expected, due to the higher frame rate and hence lower quantization error. Results are shown for only one fold, due to the high computational costs of training and evaluation at higher frame rates.

An interesting oddity in both low and high frame rate cases for Configuration II, is the performance increase when using f_c to train and f_a to obtain the ground truth for evaluation. This indicates a problem with the ground truth alignment, likely due to MIDI clock drift or similar issues, and will need to be addressed in future work. It has no effect on the main contribution of this work, which is to demonstrate that convolutional neural networks are highly sensitive to even small amounts of label noise, and to increase awareness that this issue needs to be addressed for fine grained audio signal labeling tasks.

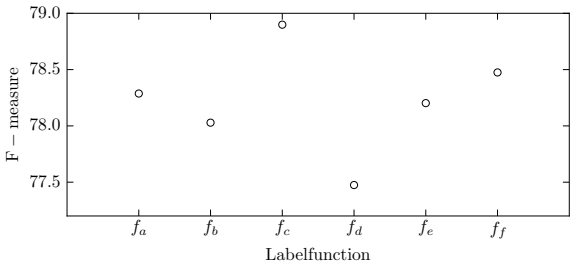


Fig. 6: Configuration II: F-measure on the test set at a frame rate of 100 [fps] across different labeling functions.

6. CONCLUSION

The effect of systematic and random label noise stemming from small misalignments of label annotations on convolutional neural networks in the context of audio signal labeling was investigated empirically, and shown to be non-negligible. We therefore conclude that great care must be taken to make sure the ground truth annotations align with the events in the audio as much as possible, especially if the intention is to use the data for fine grained sequence labeling, for example for subsequent analysis of musical timing. We demonstrated the effect with the help of already very precise annotations stemming from a mechanistic generator. We surmise that even in the case of having multiple annotations of human origin, and hence the potential to use annotator disagreement to pinpoint problematic labels and so obtain better estimates of the true label alignment, the sensitivity issue will need to be addressed carefully.

7. ACKNOWLEDGMENTS

A Tesla K40 used for parts of this research was donated by the NVIDIA Corporation. We also thank the authors and maintainers of the Lasagne [8] and Theano [9] software packages.

8. REFERENCES

[1] Devansh Arpit, Stanislaw K. Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien, "A closer look at memorization in deep networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 233–242.

[2] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, "Understanding deep learning requires rethinking generalization," *CoRR*, vol. abs/1611.03530, 2016.

[3] Rainer Kelz, Matthias Dorfer, Filip Korzeniowski, Sebastian Böck, Andreas Arzt, and Gerhard Widmer, "On the potential of simple framewise approaches to piano transcription," in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, 2016, pp. 475–481.

[4] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 24, no. 5, pp. 927–939, 2016.

[5] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *CoRR*, vol. abs/1609.04836, 2016.

[6] Valentin Emiya, Roland Badeau, and Bertrand David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. Audio, Speech & Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[7] Mert Bay, Andreas F. Ehmann, and J. Stephen Downie, "Evaluation of multiple-f0 estimation and tracking systems," in *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe International Conference Center, Kobe, Japan, October 26-30, 2009*, 2009, pp. 315–320.

[8] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, Jeffrey De Fauw, Michael Heilman, diogo149, Brian McFee, Hendrik Weideman, takacsg84, peterderivaz, Jon, instagibbs, Dr. Kashif Rasul, CongLiu, Britefury, and Jonas Degrave, "Lasagne: First release.," Aug. 2015.

[9] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio, "Theano: a CPU and GPU Math Expression Compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.