Friedrich-Alexander-Universität Erlangen-Nürnberg

Friedrich-Alexander-Universität
Erlangen-Nürnberg

Internship Report

Report Title
Training with Unaligned Dataset: Soft Dynamic Time Warping

submitted by

Quang Hoang Nguyen Vo

submitted

September 8, 2025

Supervisors

Msc. Johannes Zeitler
Prof. Dr. Meinard Müller

AUDIO
LABS

International Audio Laboratories Erlangen
A joint institution of the
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and
the Fraunhofer-Institut für Integrierte Schaltungen IIS.

Fraunhofer
IIS

Abstract

The evolution of Deep Neural Networks (DNNs) has shifted the paradigm of music information retrieval (MIR) from heuristic and mathematical models to data-driven approaches, which rely on large amounts of labelled training data. However, it introduces challenges when training with weakly aligned datasets. In this project, we investigate the characteristics of differential dynamic time warping (dDTW) through the soft-DTW (sDTW) algorithm when training with weakly aligned data. The main objective is to integrate soft-DTW as a loss function in the training process of a template-based chord recognition model. The dataset will have its chord label timestamps distorted or removed to simulate weakly or unaligned data. The dDTW loss function will then be used to train the model with the distorted dataset. The results will be compared with those obtained using the original dataset and the Connectionist Temporal Classification (CTC) loss function. Additional tasks may include experimenting and evaluating the performance of sDTW with different stablizing strategies.

Contents

Chapter 1

Introduction

Amidst the rapid advancement of deep neural networks (DNNs), the field of music information retrieval (MIR) has witnessed a significant shift from traditional heuristic and mathematical models to data-driven approaches that heavily rely on large amounts of labelled training data, such as pitch estimation [1], audio embeddings [2], automatic music transcription [3].

However, the reliance on large and accurate datasets poses many challenges, considering the time-consuming and labor-intensive nature of manual annotation, as well as the potential for human error and subjectivity. Thus, it is generally difficult to obtain strongly aligned annotations, where each frame of the audio signal is associated with a corresponding label. Instead, weakly aligned or unaligned annotations are more common, where only the presence or absence of certain labels is known, without precise temporal alignment. This ease up the data acquisition process, but requires a more sophisticated loss function to train the model. One proposed solution is using connectionist temporal classification (CTC) loss [4].

Chapter 2

Soft Dynamic Time Warping Algorithm

In this chapter, we present the mathematical formulation of the soft-DTW algorithm.

2.1 Definition and Notation

Let $\mathbf{x} = (x_0, \dots, x_{N-1}) \in \mathbb{R}^N$ and $\mathbf{y} = (y_0, \dots, y_{M-1}) \in \mathbb{R}^M$ be two time series of representing the sequence of predictions and strong targets, respectively. We then denote the soft target sequence as $\mathbf{y}' = (y'_0, \dots, y'_{M'-1}) \in \mathbb{R}^{M'}$, where $M' < M$. The objective of soft-DTW is to calculate the alignment cost matrix between $\mathbf{x}$ and $\mathbf{y}'$.

2.2 Forward Pass

In the forward pass, we compute the accumulated cost matrix $\mathbf{D}(M, N) \in \mathbb{R}^{M \times N}$, where each element $D(i, j)$ represents the minimum cost of aligning the first i elements of $\mathbf{y}'$ with the first j elements of $\mathbf{x}$. The accumulated cost is computed using the local cost matrix $\mathbf{C}$, where $\mathbf{C}(i, j) = c(x_i, y'_j)$, which measures the dissimilarity between the elements x_i and y'_j. A common choice for the local cost function is the squared Euclidean distance:

$$\mathbf{C}(i, j) = \|x_i - y'_j\|^2 \tag{2.1}$$

Instead of using the hard minimum operator as in traditional DTW, soft-DTW employs a differentiable approximation, defined as:

$$\min{}^{\gamma} = -\gamma \log \sum_{s \in S} \left(e^{-s/\gamma} \right) \tag{2.2}$$

where $\gamma > 0$ is a smoothing parameter that controls the softness of the minimum operation, and S is the set of values over which the soft minimum is computed. Combining with the step constraint of DTW, we only consider three possible predecessor cells for each cell (i, j), which

Algorithm 1 Forward Pass of Soft-DTW

```
Require: Time series $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{y}' \in \mathbb{R}^{M'}$, smoothing parameter $\gamma > 0$
Ensure: Accumulated cost matrix $\mathbf{D} \in \mathbb{R}^{M' \times N}$
  Initialize $\mathbf{D}(0, 0) = 0$
  for $i = 1$ to $M'$ do
      $\mathbf{D}(i, 0) = \infty$
  end for
  for $j = 1$ to $N$ do
      $\mathbf{D}(0, j) = \infty$
  end for
  for $i = 1$ to $M'$ do
      for $j = 1$ to $N$ do
          Compute local cost: $\mathbf{C}(i, j) = \|x_j - y'_i\|^2$
          Update accumulated cost:
              $\mathbf{D}(i, j) = \mathbf{C}(i, j) + \min^\gamma (\mathbf{D}(i-1, j), \mathbf{D}(i, j-1), \mathbf{D}(i-1, j-1))$
      end for
  end for
        return D
```

are $(i-1, j)$, $(i, j-1)$, and $(i-1, j-1)$. Thus the accumulated cost matrix $\mathbf{D}$ is computed recursively as follows:

$$\mathbf{D}(i, j) = \mathbf{C}(i, j) + \min^\gamma (\mathbf{D}(i-1, j), \mathbf{D}(i, j-1), \mathbf{D}(i-1, j-1)) \tag{2.3}$$

The recursive algorithm is summarized in Algorithm 1.

2.3 Backward Pass

In order to train the model, we need to compute the gradients of the soft-DTW loss with respect to the model parameters. The backward pass computes the gradient $H \in \mathbb{R}^{M' \times N}$ of the soft-DTW cost with respect to the cost matrix $\mathbf{C} \in \mathbb{R}^{M' \times N}$. An efficient way to compute the gradient is to use a dynamic programming approach similar to the forward pass. The gradient is computed recursively as follows:

$$\mathbf{H}(i, j) = \frac{\partial \mathbf{D}(M', N)}{\partial \mathbf{D}(i, j)} \frac{\partial \mathbf{D}(i, j)}{\partial \mathbf{C}(i, j)} \tag{2.4}$$

Chapter 3

Experimental Setup

In this section, we present the objectives of our experiment, the dataset used for training alongside with the network architecture and the training process.

3.1 Dataset

For the experiment, we use the Beatles dataset retrieved from Isophonics [5], consisting of four audio recordings with respective annotations. Since the original annotations have more than 24 chord types, which would make the network too complex and beyond the scope of this project. We therefore consider the simplified version of such annotations, which reduced the number of chord types to only 24 (12 chromas with their respective major or minor variant)[6]. We split the dataset into training, validation, and test sets. For test set, a short segment of Let It Be is used, while the rest of the dataset is split into 3:1 ratio for training and validation.

We choose a sequence length of 150 samples for training and validating the model, creating 43 and 12 segments for training and validation, respectively. In case of soft alignment, we remove the adjacent repetitions in the sequence, effectively reducing its length by around 85% on average (see Figure **??**). However, this method introduce a problem where batching target sequences of different lengths is not possible. To address this issue, after reduction, we pad the sequences repeating each frame uniformly until they reach a desired length, or "soft length". After some experiments, we found that a soft length of 16 covers all of possible reduced sequences while keeping the reduction ratio high.

3.2 Model Architecture

Given the aim of this experiment is to evaluate the performance of the proposed SDTW loss function, the network architecture plays a minor role and are kept simple. Therefore, we used a simple chord recognition network (dChord) that based on the template-based chord recognition algorithm. This network consists of a single layer that acts as the chord template to predict a 24-dimensional activation vector, corresponding to 24 chord types. The network has a total of 25 trainable parameters. Table 3.1 illustrate the components of the architecture with their

Layer	Input Dimension	Output Dimension	Parameters
Log-compression	(12, 150)	(12, 150)	0
Normalization	(12, 150)	(12, 150)	0
dChord	(12, 150)	(24, 150)	25
softmax	(24, 150)	(24, 150)	0

Table 3.1. Architecture of the chord recognition network. T is the number of time frames.

respective input and output dimensions.

During training, we use Adam optimizer with a learning rate of 0.01.

3.3 Results and Discussion

3.4 Baseline: Strongly Aligned Data with Binary Cross-entropy Loss

3.5 Weakly Aligned Data with Soft-DTW Loss

Chapter 4

Conclusions

Bibliography

[1] J. Kim, J. Salamon, P. Li, and J. Bello, “Crepe: A convolutional representation for pitch estimation,” pp. 161–165, 04 2018.

[2] A. L. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, “Look, listen, and learn more: Design choices for deep audio embeddings,” pp. 3852–3856, 2019.

[3] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, “Automatic music transcription: An overview,” *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural ’networks,” vol. 2006, pp. 369–376, 01 2006.

[5] C. Harte, M. B. Sandler, S. Abdallah, and E. Gómez, *Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations.* PhD thesis, London, UK, 2005.

[6] M. Müller, *Fundamentals of Music Processing – Audio, Analysis, Algorithms, Applications.* Springer Verlag, 2015.