INTERNATIONAL AUDIO LABORATORIES ERLANGEN

AUDIO LABS

Lecture
Selected Topics in Deep Learning
for Audio, Speech, and Music Processing

Connectionist Temporal Classification (CTC) Loss

Frank Zalkow and Meinard Müller
International Audio Laboratories Erlangen
frank.zalkow@audiolabs-erlangen.de

21.06.2021

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Fraunhofer IIS

Lecturers

Frank Zalkow

- Music Informatics and Musicology (B.A., University of Music Karlsruhe)
- Music Informatics (M.A., University of Music Karlsruhe)
- Ph.D. student in music information retrieval (FAU, supervisor: Meinard Müller)

Meinard Müller

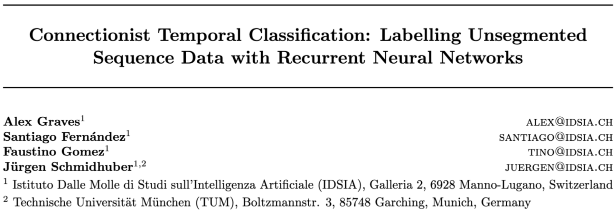
- Mathematics (Diplom/Master, Bonn University)
- Computer Science (Ph.D., Bonn University)
- Information Retrieval (Habilitation, Bonn University)
- Combinatorics (Postdoc, Keio University, Japan)
- Senior Researcher (Max-Planck Institute, Saarland)
- Professor: Semantic Audio Processing (FAU)

Overview

1. Introduction
2. CTC Loss Computation
3. Applications
4. Outlook and Further Notes

Introduction

- Connectionist Temporal Classification
- Graves, Fernández, Gomez, and Schmidhuber. *Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks.* ICML 2006. [ACM]
- "Temporal Classification": Labelling un-segmented data sequences
- "Connectionist": Refers to the use of deep learning



Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks

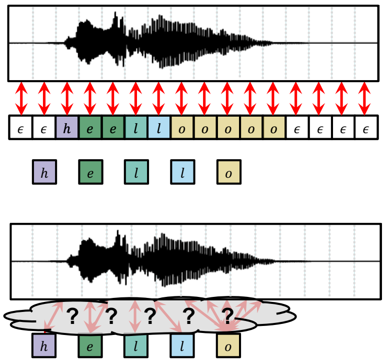
Alex Graves[1]	ALEX@IDSIA.CH
Santiago Fernández[1]	SANTIAGO@IDSIA.CH
Faustino Gomez[1]	TINO@IDSIA.CH
Jürgen Schmidhuber[1,2]	JUERGEN@IDSIA.CH

[1] Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, 6928 Manno-Lugano, Switzerland
[2] Technische Universität München (TUM), Boltzmannstr. 3, 85748 Garching, Munich, Germany

Introduction

Training data in speech recognition

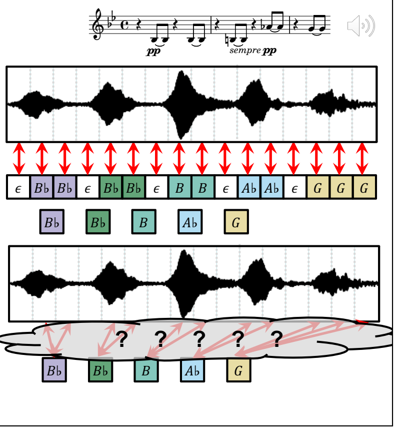
- Strongly aligned training data
  - Character annotations (labels) for each time step
  - Can be used for training in a standard classification setup
  - Tedious to annotate
- Weakly aligned training data
  - Globally corresponding character sequence without local alignment
  - Cannot be used for training in a standard classification setup
  - Easier to annotate
- Aim of CTC: Employing weakly aligned data for training
- Useful for many applications

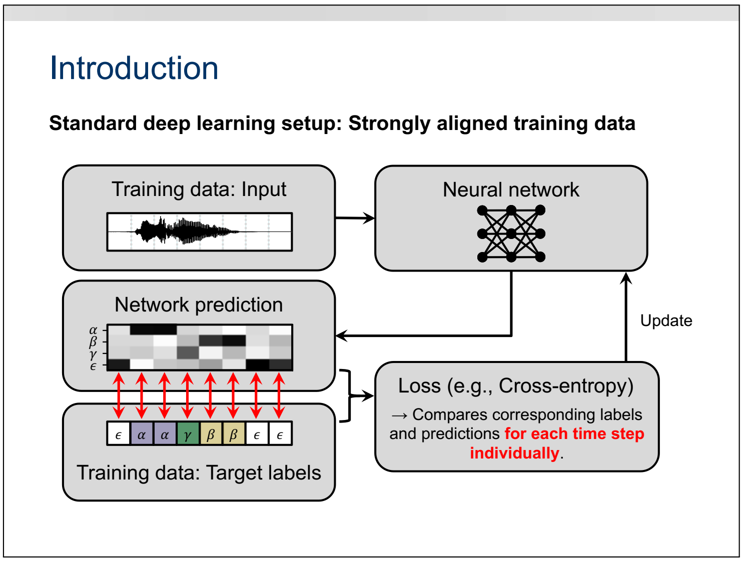


Introduction

Training data in theme-based music retrieval

- Strongly aligned training data
  - Pitch/chroma annotations (labels) for each time step
  - Can be used for training in a standard classification setup
  - Tedious to annotate
- Weakly aligned training data
  - Globally corresponding pitch/chroma sequence without local alignment
  - Cannot be used for training in a standard classification setup
  - Easier to annotate
- Aim of CTC: Employing weakly aligned data for training
- Useful for many applications





Introduction

Standard deep learning setup: Strongly aligned training data

Training data: Input

Neural network

Network prediction

α β γ ϵ

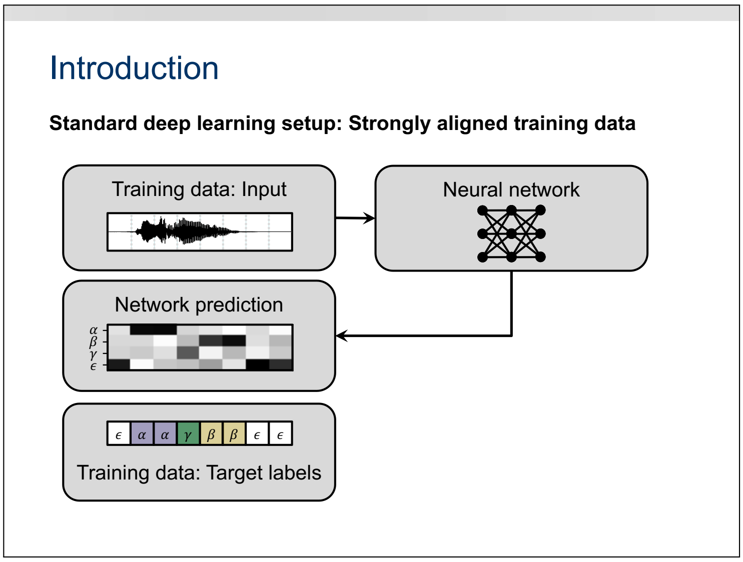
Update

ϵ α α γ β β ϵ ϵ

Training data: Target labels

Loss (e.g., Cross-entropy)

→ Compares corresponding labels and predictions **for each time step individually**.



Introduction

Standard deep learning setup: Strongly aligned training data

Training data: Input

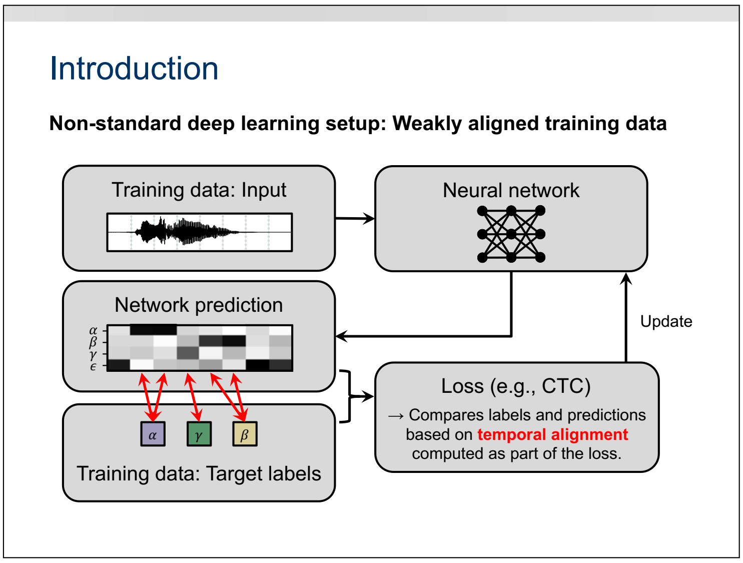
Neural network

Network prediction

α β γ ϵ

ϵ α α γ β β ϵ ϵ

Training data: Target labels



Introduction

Non-standard deep learning setup: Weakly aligned training data

Training data: Input

Neural network

Network prediction

α β γ ϵ

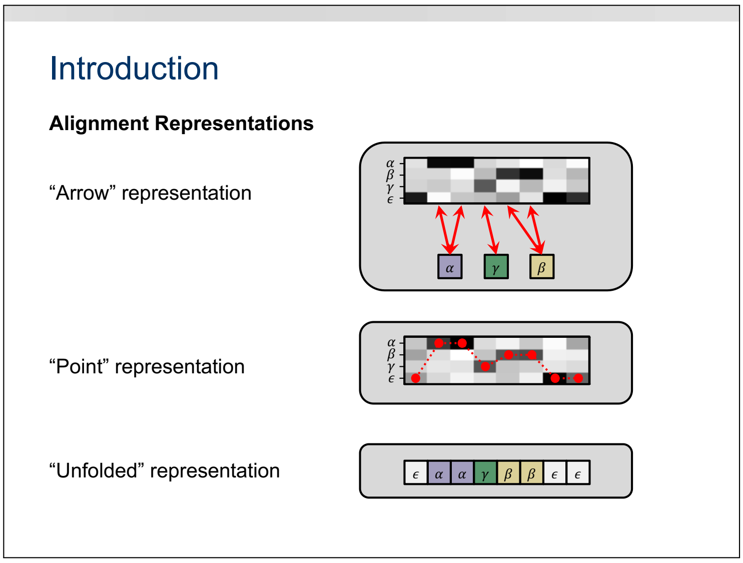
Update

α γ β

Training data: Target labels

Loss (e.g., CTC)

→ Compares labels and predictions based on **temporal alignment** computed as part of the loss.



Introduction

Alignment Representations

“Arrow” representation

α β γ ϵ

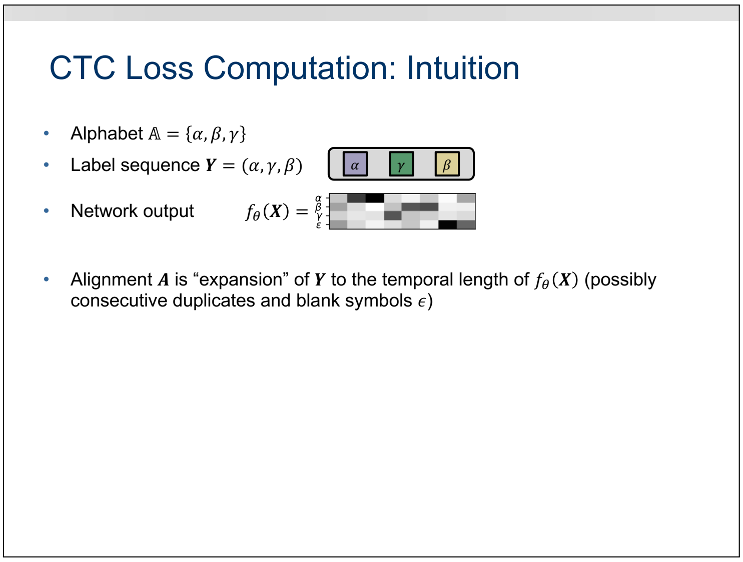
α γ β

“Point” representation

α β γ ϵ

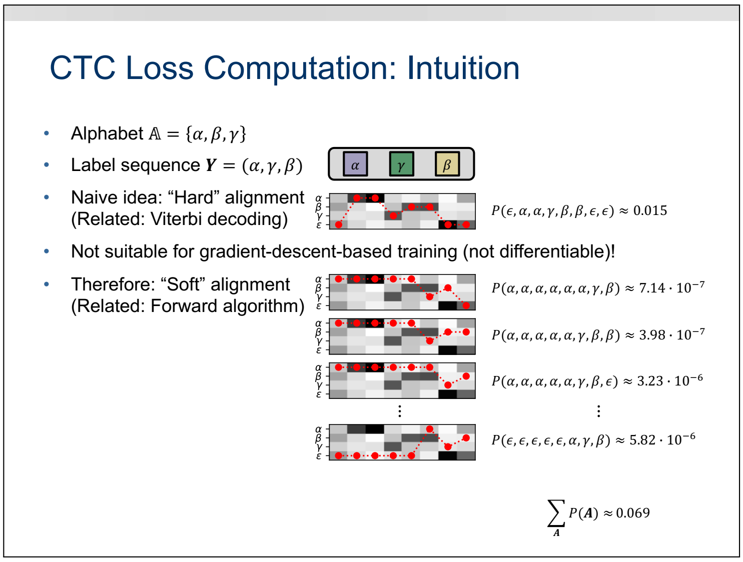
“Unfolded” representation

ϵ α α γ β β ϵ ϵ



CTC Loss Computation: Intuition

- Alphabet $\mathbb{A} = \{\alpha, \beta, \gamma\}$
- Label sequence $\boldsymbol{Y} = (\alpha, \gamma, \beta)$ α γ β
- Network output $f_\theta(\boldsymbol{X}) =$ α β γ ϵ
- Alignment $\boldsymbol{A}$ is “expansion” of $\boldsymbol{Y}$ to the temporal length of $f_\theta(\boldsymbol{X})$ (possibly consecutive duplicates and blank symbols ϵ)



CTC Loss Computation: Intuition

- Alphabet $\mathbb{A} = \{\alpha, \beta, \gamma\}$
- Label sequence $\boldsymbol{Y} = (\alpha, \gamma, \beta)$ α γ β
- Naive idea: “Hard” alignment (Related: Viterbi decoding) α β γ ϵ $P(\epsilon, \alpha, \alpha, \gamma, \beta, \beta, \epsilon, \epsilon) \approx 0.015$
- Not suitable for gradient-descent-based training (not differentiable)!
- Therefore: “Soft” alignment (Related: Forward algorithm)

α β γ ϵ $P(\alpha, \alpha, \alpha, \alpha, \alpha, \alpha, \gamma, \beta) \approx 7.14 \cdot 10^{-7}$

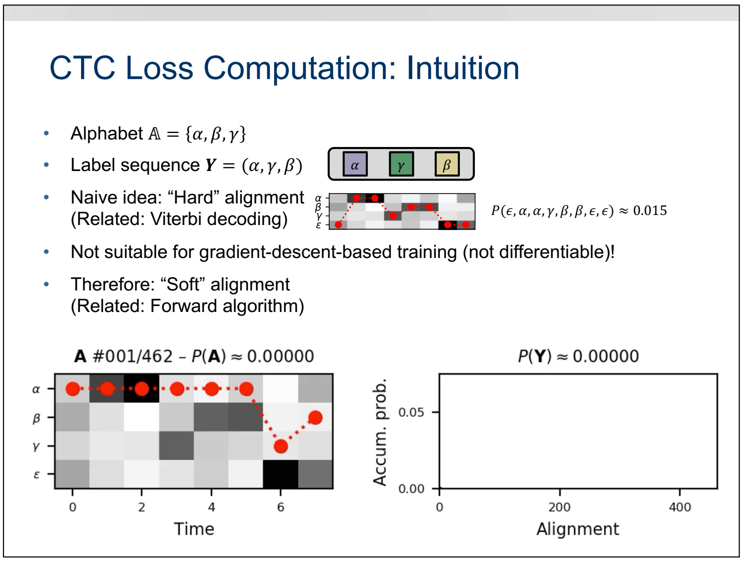
α β γ ϵ $P(\alpha, \alpha, \alpha, \alpha, \alpha, \gamma, \beta, \beta) \approx 3.98 \cdot 10^{-7}$

α β γ ϵ $P(\alpha, \alpha, \alpha, \alpha, \alpha, \gamma, \beta, \epsilon) \approx 3.23 \cdot 10^{-6}$

⋮ ⋮

α β γ ϵ $P(\epsilon, \epsilon, \epsilon, \epsilon, \epsilon, \alpha, \gamma, \beta) \approx 5.82 \cdot 10^{-6}$

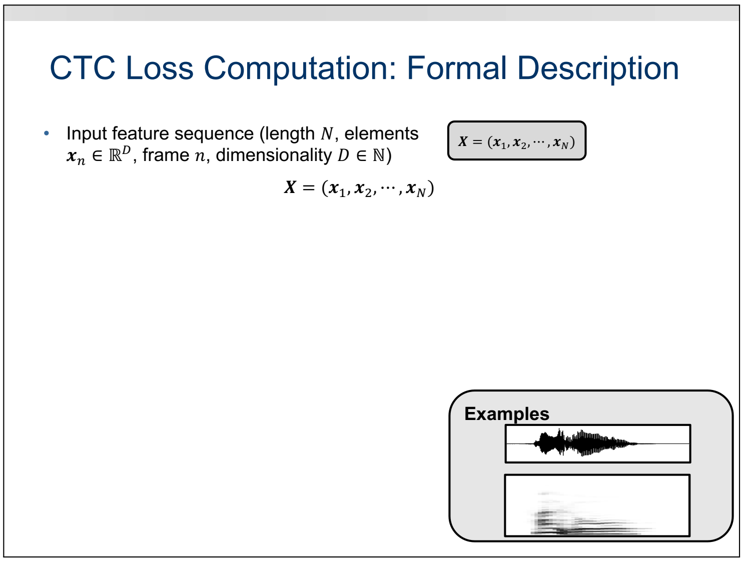
$$\sum_{\boldsymbol{A}} P(\boldsymbol{A}) \approx 0.069$$



CTC Loss Computation: Intuition

- Alphabet $\mathbb{A} = \{\alpha, \beta, \gamma\}$
- Label sequence $\boldsymbol{Y} = (\alpha, \gamma, \beta)$
- Naive idea: “Hard” alignment (Related: Viterbi decoding)
- Not suitable for gradient-descent-based training (not differentiable)!
- Therefore: “Soft” alignment (Related: Forward algorithm)

α
γ
β
$P(\epsilon, \alpha, \alpha, \gamma, \beta, \beta, \epsilon, \epsilon) \approx 0.015$
A #001/462 – P(**A**) ≈ 0.00000
α
β
γ
ϵ
0
2
4
6
Time
P(**Y**) ≈ 0.00000
Accum. prob.
0.05
0.00
0
200
400
Alignment



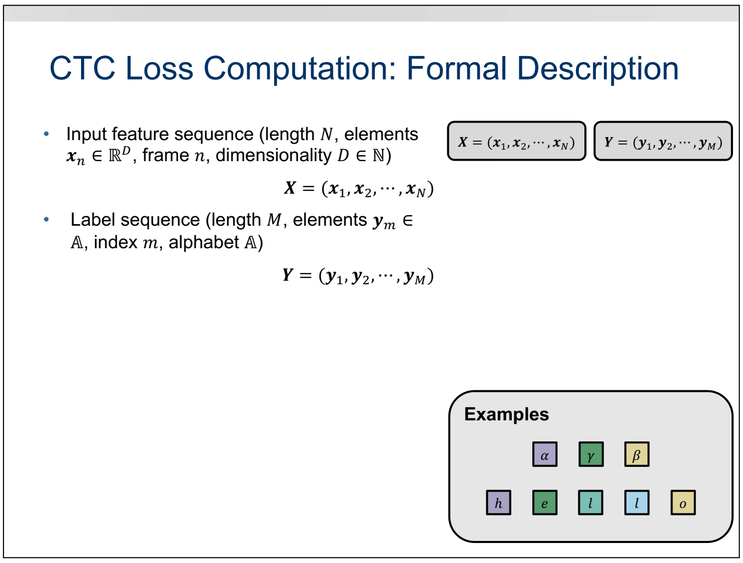
CTC Loss Computation: Formal Description

- Input feature sequence (length N, elements $\boldsymbol{x}_n \in \mathbb{R}^D$, frame n, dimensionality $D \in \mathbb{N}$)

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$$

$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$

Examples



CTC Loss Computation: Formal Description

- Input feature sequence (length N, elements $\boldsymbol{x}_n \in \mathbb{R}^D$, frame n, dimensionality $D \in \mathbb{N}$)

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$$

- Label sequence (length M, elements $\boldsymbol{y}_m \in \mathbb{A}$, index m, alphabet $\mathbb{A}$)

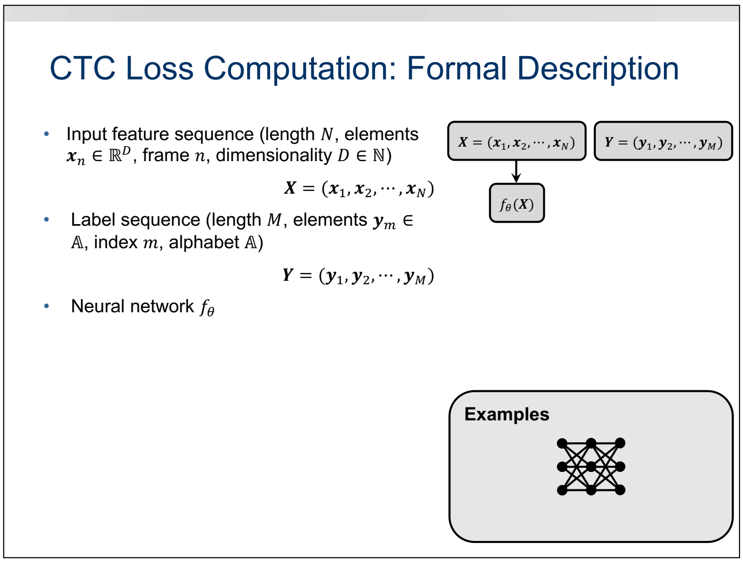
$$\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$$

$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$ $\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$

Examples

α γ β

h e l l o



CTC Loss Computation: Formal Description

- Input feature sequence (length N, elements $\boldsymbol{x}_n \in \mathbb{R}^D$, frame n, dimensionality $D \in \mathbb{N}$)

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$$

- Label sequence (length M, elements $\boldsymbol{y}_m \in \mathbb{A}$, index m, alphabet $\mathbb{A}$)

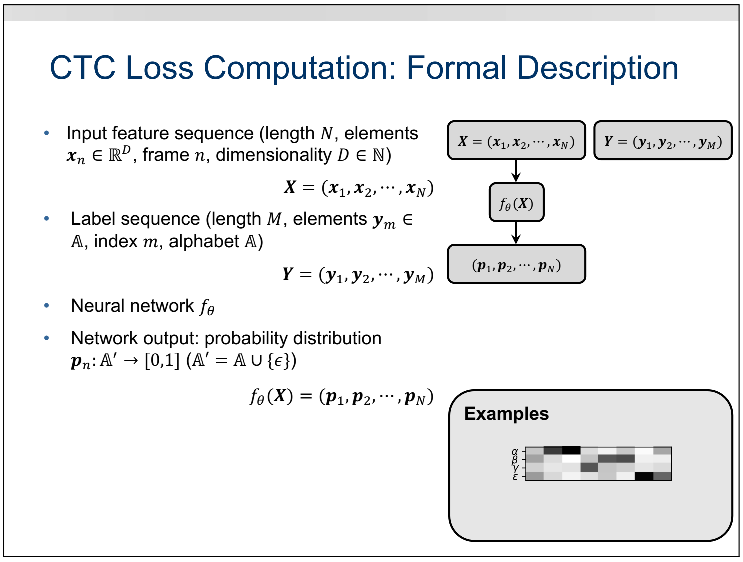
$$\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$$

- Neural network f_θ

$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$ $\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$

$f_\theta(\boldsymbol{X})$

Examples



CTC Loss Computation: Formal Description

- Input feature sequence (length N, elements $\boldsymbol{x}_n \in \mathbb{R}^D$, frame n, dimensionality $D \in \mathbb{N}$)

$$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$$

- Label sequence (length M, elements $\boldsymbol{y}_m \in \mathbb{A}$, index m, alphabet $\mathbb{A}$)

$$\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$$

- Neural network f_θ
- Network output: probability distribution $\boldsymbol{p}_n : \mathbb{A}' \to [0,1]$ ($\mathbb{A}' = \mathbb{A} \cup \{\epsilon\}$)

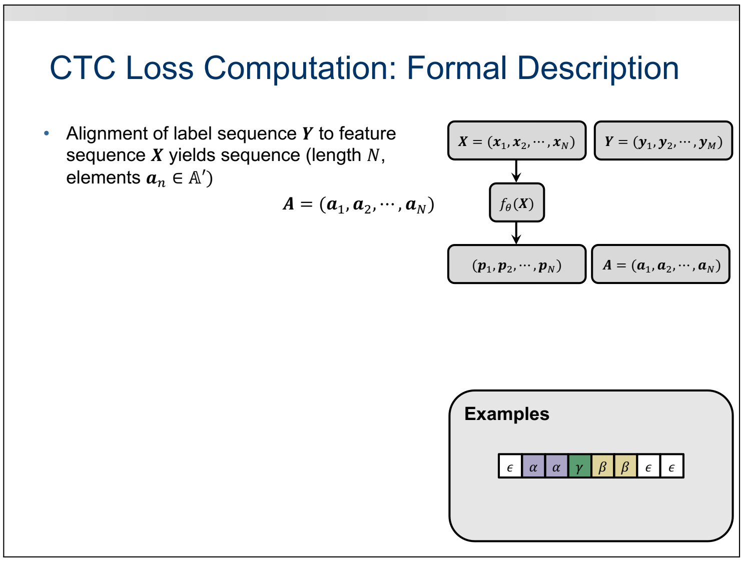
$$f_\theta(\boldsymbol{X}) = (\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N)$$

$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$ $\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$

$f_\theta(\boldsymbol{X})$

$(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N)$

Examples



CTC Loss Computation: Formal Description

- Alignment of label sequence $\boldsymbol{Y}$ to feature sequence $\boldsymbol{X}$ yields sequence (length N, elements $\boldsymbol{a}_n \in \mathbb{A}'$)

$$\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N)$$

$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$ $\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$

$f_\theta(\boldsymbol{X})$

$(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N)$ $\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N)$

Examples

ϵ α α γ β β ϵ ϵ

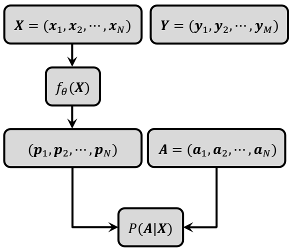
CTC Loss Computation: Formal Description

- Alignment of label sequence $\boldsymbol{Y}$ to feature sequence $\boldsymbol{X}$ yields sequence (length N, elements $\boldsymbol{a}_n \in \mathbb{A}'$)

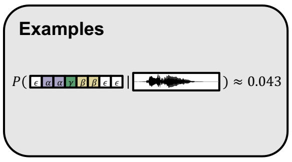
$$\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N)$$

- Probability of alignment

$$P(\boldsymbol{A}|\boldsymbol{X}) = \prod_{n=1}^{N} \boldsymbol{p}_n(\boldsymbol{a}_n)$$


$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$
$\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$
$f_\theta(\boldsymbol{X})$
$(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N)$
$\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N)$
$P(\boldsymbol{A}|\boldsymbol{X})$

Examples



CTC Loss Computation: Formal Description

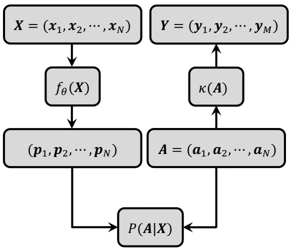
- Alignment of label sequence $\boldsymbol{Y}$ to feature sequence $\boldsymbol{X}$ yields sequence (length N, elements $\boldsymbol{a}_n \in \mathbb{A}'$)

$$\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N)$$

- Probability of alignment

$$P(\boldsymbol{A}|\boldsymbol{X}) = \prod_{n=1}^{N} \boldsymbol{p}_n(\boldsymbol{a}_n)$$

- Alignment $\boldsymbol{A}$ can be reduced to label sequence $\boldsymbol{Y}$ by composition of two functions:
  - Removing consecutive duplicates κ'
  - Removing blank symbols κ''
  - $\kappa = \kappa'' \circ \kappa'$
  - $\kappa(\boldsymbol{A}) = \kappa''\big(\kappa'(\boldsymbol{A})\big) = \boldsymbol{Y}$


$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$
$\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$
$f_\theta(\boldsymbol{X})$
$\kappa(\boldsymbol{A})$
$(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N)$
$\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N)$
$P(\boldsymbol{A}|\boldsymbol{X})$

Examples

$$\boldsymbol{A} = (\alpha, \alpha, \beta, \beta, \beta)$$
$$\kappa'(\boldsymbol{A}) = \kappa(\boldsymbol{A}) = (\alpha, \beta)$$

$$\boldsymbol{A} = (\alpha, \epsilon, \alpha, \beta, \epsilon, \beta, \beta)$$
$$\kappa'(\boldsymbol{A}) = (\alpha, \epsilon, \alpha, \beta, \epsilon, \beta)$$
$$\kappa''\big(\kappa'(\boldsymbol{A})\big) = \kappa(\boldsymbol{A}) = (\alpha, \alpha, \beta, \beta)$$

CTC Loss Computation: Formal Description

- Many different alignments $\boldsymbol{A}$ map to the same label sequence $\boldsymbol{Y}$ (“valid alignments”)
- Set of all valid alignments

$$\mathbb{K}_{\boldsymbol{X},\boldsymbol{Y}} = \{\boldsymbol{A} \in (\mathbb{A}')^N : \kappa(\boldsymbol{A}) = \boldsymbol{Y}\}$$

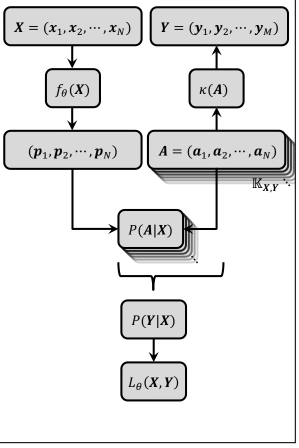
- Probability of label sequence

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{\boldsymbol{A} \in \mathbb{K}_{\boldsymbol{X},\boldsymbol{Y}}} P(\boldsymbol{A}|\boldsymbol{X})$$

- CTC loss

$$L_\theta(\boldsymbol{X}, \boldsymbol{Y}) = -\log P(\boldsymbol{Y}|\boldsymbol{X})$$

- Problem: Combinatorial explosion in the cardinality of the set $\mathbb{K}_{\boldsymbol{X},\boldsymbol{Y}}$
- Solution: Compute $P(\boldsymbol{Y}|\boldsymbol{X})$ by dynamic programing


$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$
$\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$
$f_\theta(\boldsymbol{X})$
$\kappa(\boldsymbol{A})$
$(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N)$
$\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N)$
$\mathbb{K}_{\boldsymbol{X},\boldsymbol{Y}}$
$P(\boldsymbol{A}|\boldsymbol{X})$
$P(\boldsymbol{Y}|\boldsymbol{X})$
$L_\theta(\boldsymbol{X}, \boldsymbol{Y})$

CTC Loss Computation: Formal Description

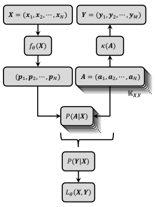
- Modified label sequence (length $2M + 1$, elements $\boldsymbol{y}_m \in \mathbb{A}'$ for $m \in \{1, \cdots, 2M + 1\}$)

$$\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_{2M+1})$$

  obtained by inserting blank symbols ϵ into $\boldsymbol{Y}$ at the beginning, the end, and between each two consecutive elements

- Idea: Align network output $(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N)$ to modified label sequence $(\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_{2M+1})$, but allow for skipping the inserted blank symbols ϵ (except where it is used to separate consecutive duplicates)
- Set of all alignments that correspond to $\boldsymbol{Z}$

$$\mathbb{K}'_{\boldsymbol{X},\boldsymbol{Z}} = \{\boldsymbol{A} \in (\mathbb{A}')^N : \kappa'(\boldsymbol{A}) = \boldsymbol{Z}\}$$


$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$
$\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$
$f_\theta(\boldsymbol{X})$
$\kappa(\boldsymbol{A})$
$(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N)$
$\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N)$
$\mathbb{K}_{\boldsymbol{X},\boldsymbol{Y}}$
$P(\boldsymbol{A}|\boldsymbol{X})$
$P(\boldsymbol{Y}|\boldsymbol{X})$
$L_\theta(\boldsymbol{X}, \boldsymbol{Y})$

Examples

$$\boldsymbol{Y} = (\alpha, \beta)$$
$$\boldsymbol{Z} = (\epsilon, \alpha, \epsilon, \beta, \epsilon)$$

$$\boldsymbol{Y} = (\alpha, \alpha, \beta, \beta, \beta)$$
$$\boldsymbol{Z} = (\epsilon, \alpha, \epsilon, \alpha, \epsilon, \beta, \epsilon, \beta, \epsilon, \beta, \epsilon)$$

CTC Loss Computation: Formal Description

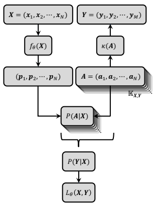
- Forward matrix

$$\boldsymbol{D}(m, n) = \sum_{\substack{\boldsymbol{A} \in \mathbb{K}'_{\boldsymbol{X},\boldsymbol{Z}} \\ \kappa'(\boldsymbol{A}(1:n)) = \boldsymbol{Z}(1:m)}} \prod_{i=1}^{n} \boldsymbol{p}_i(\boldsymbol{a}_i)$$

  of $n \in \{1, \cdots, N\}$ and $m \in \{1, \cdots, 2M + 1\}$, encoding the probability that the first n time steps correspond to the first m elements of the modified label sequence $\boldsymbol{Z}$

- Forward matrix contains probability of label sequence

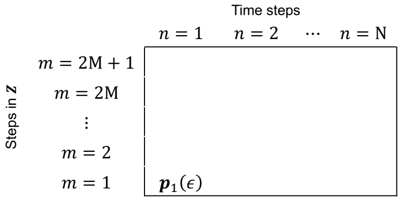
$$P(\boldsymbol{Y}|\boldsymbol{X}) = \boldsymbol{D}(2M + 1, N) + \boldsymbol{D}(2M, N)$$


$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$
$\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_M)$
$f_\theta(\boldsymbol{X})$
$\kappa(\boldsymbol{A})$
$(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_N)$
$\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N)$
$\mathbb{K}_{\boldsymbol{X},\boldsymbol{Y}}$
$P(\boldsymbol{A}|\boldsymbol{X})$
$P(\boldsymbol{Y}|\boldsymbol{X})$
$L_\theta(\boldsymbol{X}, \boldsymbol{Y})$

CTC Loss Computation: Formal Description

- Initialization

$$\boldsymbol{D}(1,1) = \boldsymbol{p}_1(\boldsymbol{z}_1) = \boldsymbol{p}_1(\epsilon)$$



CTC Loss Computation: Formal Description

- Initialization

$$\boldsymbol{D}(1,1) = \boldsymbol{p}_1(\boldsymbol{z}_1) = \boldsymbol{p}_1(\epsilon)$$
$$\boldsymbol{D}(2,1) = \boldsymbol{p}_1(\boldsymbol{z}_2) = \boldsymbol{p}_1(\boldsymbol{y}_1)$$

Time steps

Steps in $\boldsymbol{Z}$

$$\begin{array}{r|llll|}
 & n=1 & n=2 & \cdots & n=\mathrm{N} \\ \hline
m = 2\mathrm{M}+1 & & & & \\
m = 2\mathrm{M} & & & & \\
\vdots & & & & \\
m = 2 & \boldsymbol{p}_1(\boldsymbol{y}_1) & & & \\
m = 1 & \boldsymbol{p}_1(\epsilon) & & & \\ \hline
\end{array}$$

CTC Loss Computation: Formal Description

- Initialization

$$\boldsymbol{D}(1,1) = \boldsymbol{p}_1(\boldsymbol{z}_1) = \boldsymbol{p}_1(\epsilon)$$
$$\boldsymbol{D}(2,1) = \boldsymbol{p}_1(\boldsymbol{z}_2) = \boldsymbol{p}_1(\boldsymbol{y}_1)$$
$$\boldsymbol{D}(m,1) = 0, \qquad \forall m > 2$$

Time steps

Steps in $\boldsymbol{Z}$

$$\begin{array}{r|llll|}
 & n=1 & n=2 & \cdots & n=\mathrm{N} \\ \hline
m = 2\mathrm{M}+1 & 0 & & & \\
m = 2\mathrm{M} & 0 & & & \\
\vdots & 0 & & & \\
m = 2 & \boldsymbol{p}_1(\boldsymbol{y}_1) & & & \\
m = 1 & \boldsymbol{p}_1(\epsilon) & & & \\ \hline
\end{array}$$

CTC Loss Computation: Formal Description

- Initialization

$$\boldsymbol{D}(1,1) = \boldsymbol{p}_1(\boldsymbol{z}_1) = \boldsymbol{p}_1(\epsilon)$$
$$\boldsymbol{D}(2,1) = \boldsymbol{p}_1(\boldsymbol{z}_2) = \boldsymbol{p}_1(\boldsymbol{y}_1)$$
$$\boldsymbol{D}(m,1) = 0, \qquad \forall m > 2$$

- Recursion

$$\boldsymbol{D}(m,n) = \boldsymbol{p}_n(\boldsymbol{z}_m) \cdot \begin{cases} (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1)), & \text{if } \boldsymbol{z}_m = \epsilon \\ & \\ & \end{cases}$$

Time steps

Steps in $\boldsymbol{Z}$

$$\begin{array}{r|llll|}
 & n=1 & n=2 & \cdots & n=\mathrm{N} \\ \hline
m = 2\mathrm{M}+1 & 0 & & & \\
m = 2\mathrm{M} & 0 & & & \\
\vdots & 0 & & & \\
m = 2 & \boldsymbol{p}_1(\boldsymbol{y}_1) & & & \\
m = 1 & \boldsymbol{p}_1(\epsilon) & & & \\ \hline
\end{array}$$

Example

$$\boldsymbol{Z} = (\cdots, \epsilon, \alpha, \boldsymbol{\epsilon}, \alpha, \epsilon, \beta, \epsilon, \cdots)$$

CTC Loss Computation: Formal Description

- Initialization

$$\boldsymbol{D}(1,1) = \boldsymbol{p}_1(\boldsymbol{z}_1) = \boldsymbol{p}_1(\epsilon)$$
$$\boldsymbol{D}(2,1) = \boldsymbol{p}_1(\boldsymbol{z}_2) = \boldsymbol{p}_1(\boldsymbol{y}_1)$$
$$\boldsymbol{D}(m,1) = 0, \qquad \forall m > 2$$

- Recursion

$$\boldsymbol{D}(m,n) = \boldsymbol{p}_n(\boldsymbol{z}_m) \cdot \begin{cases} (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1)), & \text{if } \boldsymbol{z}_m = \epsilon \\ (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1)), & \text{if } \boldsymbol{z}_{m-2} = \boldsymbol{z}_m \\ & \end{cases}$$

Time steps

Steps in $\boldsymbol{Z}$

$$\begin{array}{r|llll|}
 & n=1 & n=2 & \cdots & n=\mathrm{N} \\ \hline
m = 2\mathrm{M}+1 & 0 & & & \\
m = 2\mathrm{M} & 0 & & & \\
\vdots & 0 & & & \\
m = 2 & \boldsymbol{p}_1(\boldsymbol{y}_1) & & & \\
m = 1 & \boldsymbol{p}_1(\epsilon) & & & \\ \hline
\end{array}$$

Example

$$\boldsymbol{Z} = (\cdots, \epsilon, \alpha, \epsilon, \boldsymbol{\alpha}, \epsilon, \beta, \epsilon, \cdots)$$

CTC Loss Computation: Formal Description

- Initialization

$$\boldsymbol{D}(1,1) = \boldsymbol{p}_1(\boldsymbol{z}_1) = \boldsymbol{p}_1(\epsilon)$$
$$\boldsymbol{D}(2,1) = \boldsymbol{p}_1(\boldsymbol{z}_2) = \boldsymbol{p}_1(\boldsymbol{y}_1)$$
$$\boldsymbol{D}(m,1) = 0, \qquad \forall m > 2$$

- Recursion

$$\boldsymbol{D}(m,n) = \boldsymbol{p}_n(\boldsymbol{z}_m) \cdot \begin{cases} (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1)), & \text{if } \boldsymbol{z}_m = \epsilon \\ (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1)), & \text{if } \boldsymbol{z}_{m-2} = \boldsymbol{z}_m \\ (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1) + \boldsymbol{D}(m-2,n-1)), & \text{otherwise} \end{cases}$$

Time steps

Steps in $\boldsymbol{Z}$

$$\begin{array}{r|llll|}
 & n=1 & n=2 & \cdots & n=\mathrm{N} \\ \hline
m = 2\mathrm{M}+1 & 0 & & & \\
m = 2\mathrm{M} & 0 & & & \\
\vdots & 0 & & & \\
m = 2 & \boldsymbol{p}_1(\boldsymbol{y}_1) & & & \\
m = 1 & \boldsymbol{p}_1(\epsilon) & & & \\ \hline
\end{array}$$

Example

$$\boldsymbol{Z} = (\cdots, \epsilon, \alpha, \epsilon, \alpha, \epsilon, \boldsymbol{\beta}, \epsilon, \cdots)$$

CTC Loss Computation: Formal Description

- Initialization

$$\boldsymbol{D}(1,1) = \boldsymbol{p}_1(\boldsymbol{z}_1) = \boldsymbol{p}_1(\epsilon)$$
$$\boldsymbol{D}(2,1) = \boldsymbol{p}_1(\boldsymbol{z}_2) = \boldsymbol{p}_1(\boldsymbol{y}_1)$$
$$\boldsymbol{D}(m,1) = 0, \qquad \forall m > 2$$

- Recursion

$$\boldsymbol{D}(m,n) = \boldsymbol{p}_n(\boldsymbol{z}_m) \cdot \begin{cases} (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1)), & \text{if } \boldsymbol{z}_m = \epsilon \\ (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1)), & \text{if } \boldsymbol{z}_{m-2} = \boldsymbol{z}_m \\ (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1) + \boldsymbol{D}(m-2,n-1)), & \text{otherwise} \end{cases}$$

Time steps

Steps in $\boldsymbol{Z}$

$$\begin{array}{r|llll|}
 & n=1 & n=2 & \cdots & n=\mathrm{N} \\ \hline
m = 2\mathrm{M}+1 & 0 & \cdots & \cdots & P(\boldsymbol{Y}|\boldsymbol{X}) \\
m = 2\mathrm{M} & 0 & \cdots & \cdots & \\
\vdots & 0 & \cdots & \cdots & \cdots \\
m = 2 & \boldsymbol{p}_1(\boldsymbol{y}_1) & \cdots & \cdots & \cdots \\
m = 1 & \boldsymbol{p}_1(\epsilon) & \cdots & \cdots & \cdots \\ \hline
\end{array}$$

CTC Loss Computation: Formal Description

- Initialization

$$\begin{aligned} \boldsymbol{D}(1,1) &= \boldsymbol{p}_1(\boldsymbol{z}_1) = \boldsymbol{p}_1(\epsilon) \\ \boldsymbol{D}(2,1) &= \boldsymbol{p}_1(\boldsymbol{z}_2) = \boldsymbol{p}_1(\boldsymbol{y}_1) \\ \boldsymbol{D}(m,1) &= 0, \qquad \forall m > 2 \end{aligned}$$

- Recursion

$$\boldsymbol{D}(m,n) = \boldsymbol{p}_n(\boldsymbol{z}_m) \cdot \begin{cases} (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1)), & \text{if } \boldsymbol{z}_m = \epsilon \\ (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1)), & \text{if } \boldsymbol{z}_{m-2} = \boldsymbol{z}_m \\ (\boldsymbol{D}(m,n-1) + \boldsymbol{D}(m-1,n-1) + \boldsymbol{D}(m-2,n-1)), & \text{otherwise} \end{cases}$$

Time steps

Steps in $\boldsymbol{Z}$

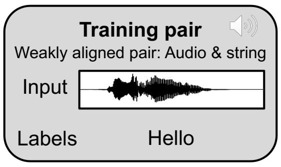
	$n = 1$	$n = 2$	$\cdots$	$n = \mathrm{N}$
$m = 2\mathrm{M}+1$	0	$\cdots$	$\cdots$	$P(\boldsymbol{Y}\vert\boldsymbol{X})$
$m = 2\mathrm{M}$	0	$\cdots$	$\cdots$	
$\vdots$	0	$\cdots$	$\cdots$	$\cdots$
$m = 2$	$\boldsymbol{p}_1(\boldsymbol{y}_1)$	$\cdots$	$\cdots$	$\cdots$
$m = 1$	$\boldsymbol{p}_1(\epsilon)$	$\cdots$	$\cdots$	$\cdots$

Difference to hard alignment: Sum instead of max!

Applications

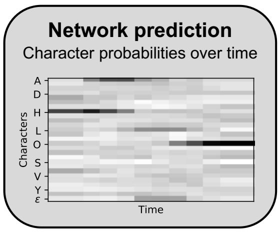
Application: Speech Recognition

- Task: Estimate character labels from waveform
- CTC-based training: Using weakly aligned pairs to train DNN for computing character probabilities[1]
- Classes: 26 characters, space, and blank symbol
- Approach: Finding most probable character sequence for given character probabilities, e.g., using beam search
- CTC is a core technology used in today's speech recognizing systems, e.g., in the Google App[2]



Training pair

Weakly aligned pair: Audio & string



Network prediction

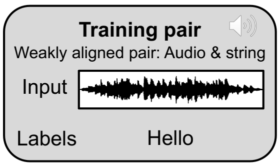
Character probabilities over time

[1] Graves et al. *Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks*. ICML 2006. [ACM]

[2] Sak et al. *Google Voice Search: Faster and More Accurate*. Google AI Blog, 2015 (https://ai.googleblog.com/2015/09/google-voice-search-faster-and-more.html).

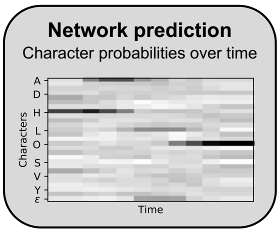
Application: Lyrics Alignment

- Task: Align character labels from lyrics to music recording[1]
- CTC-based training: Using weakly aligned pairs to train DNN for computing character probabilities (as before)[2]
- Classes: 26 characters, space, and blank symbol (as before)
- But: Now acoustic model needs to ignore non-singing-voice components of input representation
- Approach: Finding most probable alignment for given lyrics (dynamic programming similar to Viterbi)
- Beyond lyrics alignment, lyrics transcription is a challenging problem[1]



Training pair

Weakly aligned pair: Audio & string



Network prediction

Character probabilities over time

[1] Stoller et al. *End-to-end Lyrics Alignment for Polyphonic Music Using an Audio-To-Character Recognition Model*. ICASSP 2019. [IEEE]

[2] Graves et al. *Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks*. ICML 2006. [ACM]

Demo: Synced lyrics feature in the *Apple Music* App

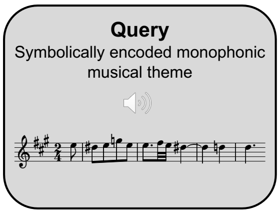
Song: *Jingle Bells* by Pentatonix

Source: Twitter, @PTXofficial
https://twitter.com/PTXofficial/status/1337556809553301506

Sorry for the untimely Christmas music!

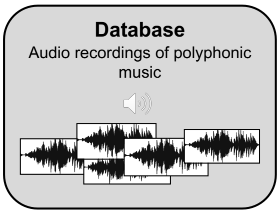
Application: Theme-Based Music Retrieval

- Task: Given a symbolically encoded musical theme, find music recordings, where theme is being played[1–3]
- Challenges due to differences in:[1]
  - Modality (symbolic vs. audio)
  - Tuning
  - Transposition
  - Tempo
  - Degree of polyphony



Query

Symbolically encoded monophonic musical theme



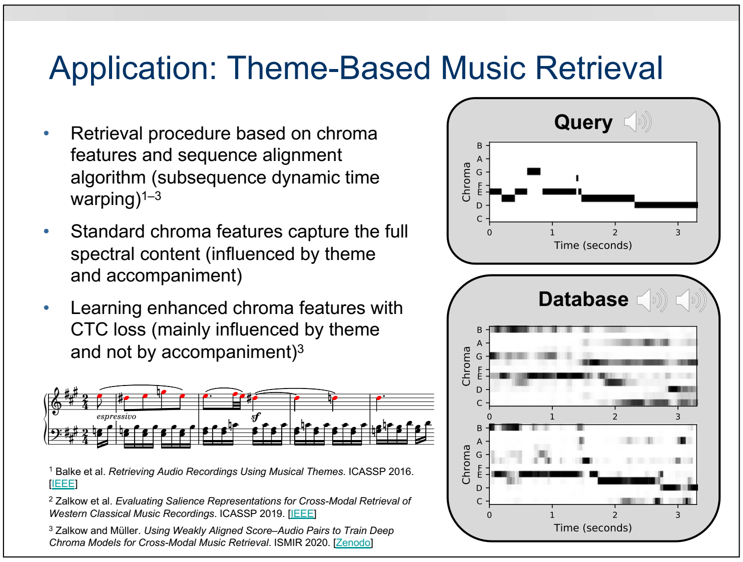
Database

Audio recordings of polyphonic music

[1] Balke et al. *Retrieving Audio Recordings Using Musical Themes*. ICASSP 2016. [IEEE]

[2] Zalkow et al. *Evaluating Salience Representations for Cross-Modal Retrieval of Western Classical Music Recordings*. ICASSP 2019. [IEEE]

[3] Zalkow and Müller. *Using Weakly Aligned Score–Audio Pairs to Train Deep Chroma Models for Cross-Modal Music Retrieval*. ISMIR 2020. [Zenodo]



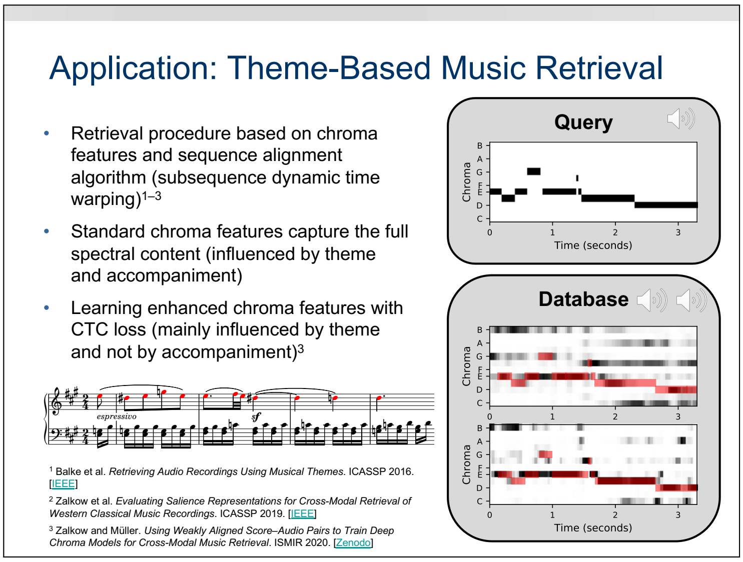
Application: Theme-Based Music Retrieval

- Retrieval procedure based on chroma features and sequence alignment algorithm (subsequence dynamic time warping)[1–3]
- Standard chroma features capture the full spectral content (influenced by theme and accompaniment)
- Learning enhanced chroma features with CTC loss (mainly influenced by theme and not by accompaniment)[3]

[1] Balke et al. *Retrieving Audio Recordings Using Musical Themes.* ICASSP 2016. [IEEE]

[2] Zalkow et al. *Evaluating Salience Representations for Cross-Modal Retrieval of Western Classical Music Recordings.* ICASSP 2019. [IEEE]

[3] Zalkow and Müller. *Using Weakly Aligned Score–Audio Pairs to Train Deep Chroma Models for Cross-Modal Music Retrieval.* ISMIR 2020. [Zenodo]



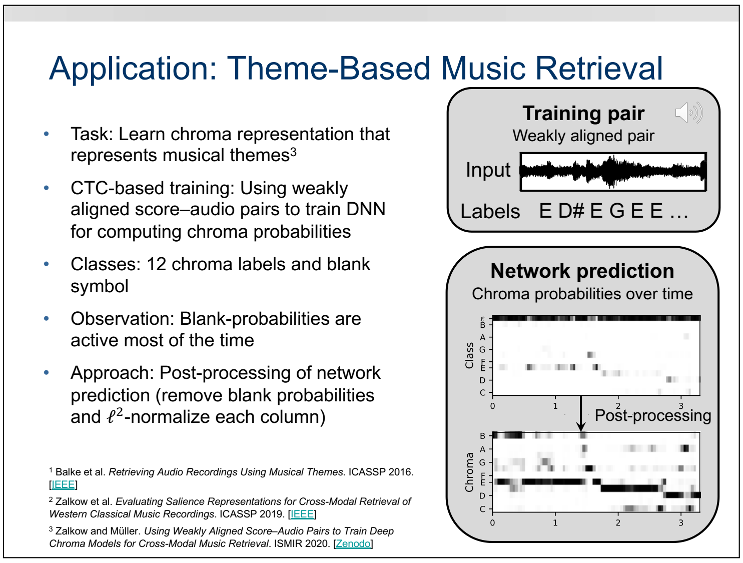
Application: Theme-Based Music Retrieval

- Retrieval procedure based on chroma features and sequence alignment algorithm (subsequence dynamic time warping)[1–3]
- Standard chroma features capture the full spectral content (influenced by theme and accompaniment)
- Learning enhanced chroma features with CTC loss (mainly influenced by theme and not by accompaniment)[3]

[1] Balke et al. *Retrieving Audio Recordings Using Musical Themes.* ICASSP 2016. [IEEE]

[2] Zalkow et al. *Evaluating Salience Representations for Cross-Modal Retrieval of Western Classical Music Recordings.* ICASSP 2019. [IEEE]

[3] Zalkow and Müller. *Using Weakly Aligned Score–Audio Pairs to Train Deep Chroma Models for Cross-Modal Music Retrieval.* ISMIR 2020. [Zenodo]



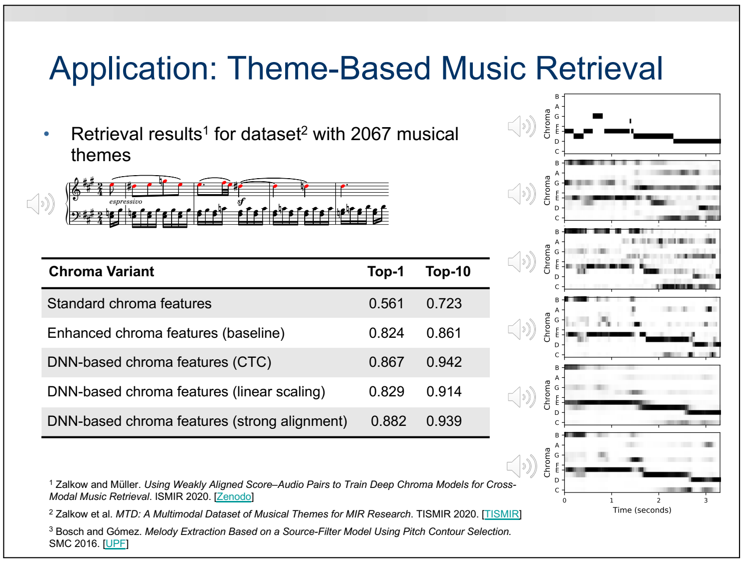
Application: Theme-Based Music Retrieval

- Task: Learn chroma representation that represents musical themes[3]
- CTC-based training: Using weakly aligned score–audio pairs to train DNN for computing chroma probabilities
- Classes: 12 chroma labels and blank symbol
- Observation: Blank-probabilities are active most of the time
- Approach: Post-processing of network prediction (remove blank probabilities and ℓ^2-normalize each column)

[1] Balke et al. *Retrieving Audio Recordings Using Musical Themes.* ICASSP 2016. [IEEE]

[2] Zalkow et al. *Evaluating Salience Representations for Cross-Modal Retrieval of Western Classical Music Recordings.* ICASSP 2019. [IEEE]

[3] Zalkow and Müller. *Using Weakly Aligned Score–Audio Pairs to Train Deep Chroma Models for Cross-Modal Music Retrieval.* ISMIR 2020. [Zenodo]



Application: Theme-Based Music Retrieval

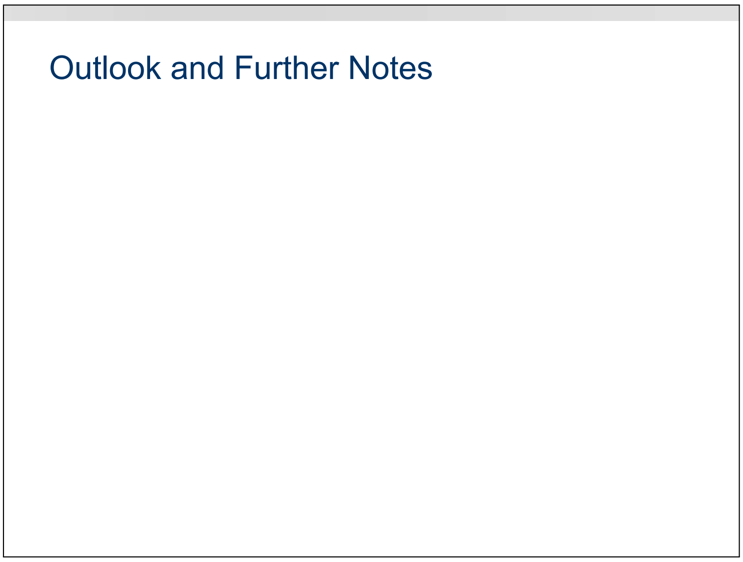
- Retrieval results[1] for dataset[2] with 2067 musical themes

Chroma Variant	Top-1	Top-10
Standard chroma features	0.561	0.723
Enhanced chroma features (baseline)	0.824	0.861
DNN-based chroma features (CTC)	0.867	0.942
DNN-based chroma features (linear scaling)	0.829	0.914
DNN-based chroma features (strong alignment)	0.882	0.939

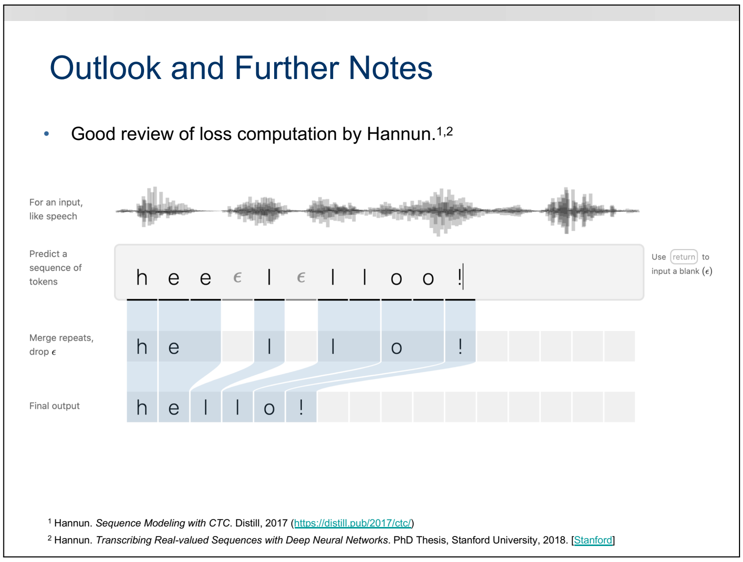
[1] Zalkow and Müller. *Using Weakly Aligned Score–Audio Pairs to Train Deep Chroma Models for Cross-Modal Music Retrieval.* ISMIR 2020. [Zenodo]

[2] Zalkow et al. *MTD: A Multimodal Dataset of Musical Themes for MIR Research.* TISMIR 2020. [TISMIR]

[3] Bosch and Gómez. *Melody Extraction Based on a Source-Filter Model Using Pitch Contour Selection.* SMC 2016. [UPF]



Outlook and Further Notes



Outlook and Further Notes

- Good review of loss computation by Hannun.[1,2]

For an input, like speech
Predict a sequence of tokens
h e e ϵ l ϵ l l o o !
Use return to input a blank (ϵ)
Merge repeats, drop ϵ
h e l l o !
Final output
h e l l o !

[1] Hannun. *Sequence Modeling with CTC.* Distill, 2017 (https://distill.pub/2017/ctc/)

[2] Hannun. *Transcribing Real-valued Sequences with Deep Neural Networks.* PhD Thesis, Stanford University, 2018. [Stanford]

Outlook and Further Notes

- CTC is suited for gradient-based training because it accumulates all possible alignments (i.e., summation over the set $\mathbb{K}_{X,Y}$). This procedure is different from related alignment algorithms such as dynamic time warping (DTW), where the optimal alignment is computed (corresponding to a maximization over $\mathbb{K}_{X,Y}$). A variant of DTW adapted for the usage with neural networks is known as soft-DTW.[1]

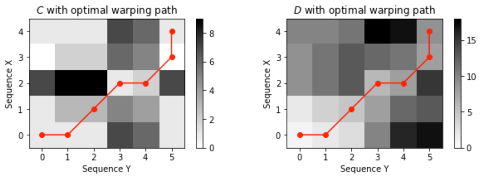


Figure from FMP[2]

[1] Cuturi and Blondel: *Soft-DTW: A Differentiable Loss Function for Time-Series*, ICML 2017. [PMLR]

[2] Müller and Zalkow. *FMP Notebooks: Educational Material for Teaching and Learning Fundamentals of Music Processing*. ISMIR 2019. [Zenodo]

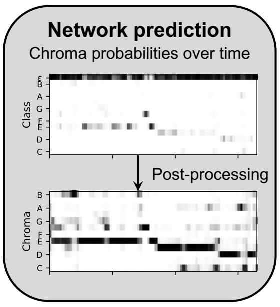
Outlook and Further Notes

- A CTC-based network models the dependencies between input and output. However, the dependencies between individual output elements are not modeled explicitly. One may use an external “language model” if needed. An extension of CTC for jointly modeling both input–output and output–output dependencies is denoted as Sequence Transduction.[1]

[1] Graves: *Sequence Transduction with Recurrent Neural Networks*. ICML Workshop on Representation Learning 2012. [arXiv]

Outlook and Further Notes

- Many studies show that the probabilities for the blank symbol are often dominant in the output of CTC-based networks (“spiky problem”). One may modify the CTC algorithm to compensate for that.[1]



Network prediction

Chroma probabilities over time

Post-processing

[1] Li and Wang: Reinterpreting CTC Training as Iterative Fitting. Pattern Recognition, 2020. [ScienceDirect, arXiv]

Outlook and Further Notes

- In its basic form, CTC is only able to model one label at a time (music application: “monophonic”). There are also extensions of CTC to multi-label problems (music application: “polyphonic”).[1]

[1] Wigington et al. *Multi-Label Connectionist Temporal Classification*. ICDAR 2019. [IEEE]