香川高専

卒業研究論文

論 文 題 目

深層学習を用いた単一話者発話区間検出

提出年月日	令和 6 年 2 月 26 日	
学　　科	電 気 情 報 工 学 科	
氏　　名	加藤 大輝	印
指導教員（主査）	北村 大地 講師	印
副　　査	雛元 洋一 助教	印
学 科 長	漆原 史郎 教授	印

香川高等専門学校

Single Voice Activity Detection Using Deep Learning

Taiki Kato

Department of Electrical and Computer Engineering
National Institute of Technology, Kagawa College

Abstract

Blind source separation (BSS) is a technique that aims to estimate individual source signals from observed signals without knowing speakers' information. This technique is utilized for various applications such as background noise suppression in hearing-aid systems, separated speech estimation in a conversation, and separation of each musical instrument sound. Independent vector analysis (IVA) is a well-known BSS technique, and it has been confirmed that the BSS separation accuracy of IVA is improved when the mixture signal contains many single-voice active segments. It is believed that using only single-voice active segments as an observed signal can enhance the separation accuracy of IVA. To achieve such performance improvement, a new technique that accurately estimates the single-voice active segments from a multi-speaker audio signal is required. In this thesis, I define this problem as single-voice activity detection (SVAD). I propose a supervised approach using a deep neural network (DNN) for SVAD. The proposed method is based on a recurrent neural network with long short-term memory units, which can efficiently train the temporal structure of observed speech mixture signals for SVAD. To train the proposed method, a dataset consisting of Japanese male and female speakers is prepared, and the labels for single-voice activity segments are calculated. Experimental results under various conditions show that the proposed method achieves approximately 90% accuracy in detecting single-voice active segments in test data.

Keywords: blind source separation, voice activity detection, deep neural networks

（和訳）

ブラインド音源分離（blind source separation: BSS）とは，話者情報が未知の音源信号が混ざった観測信号から，混ざる前の個々の音源を推定する技術である．この技術は補聴器などの背景雑音の抑圧，複数人の会話音声から各話者の音声の推定，音響信号から各楽器音の推定を行うことなどに利用される．BSS の有名な技術である独立ベクトル分析（independent vector analysis: IVA）では混合音声信号内に一人だけが発話している時間区間（以後，単一話者発話区間と呼ぶ）が多いほど BSS の分離精度向上に寄与されることが確認されている．従って，観測信号として単一話者発話区間のみを用いることで，IVA の分離精度を向上できると考えられる．このような BSS の性能向上を実現するためには，複数の話者の音声が混合している音響信号から単一話者発話区間を正確に検出する技術が必要である．そこで本論文では，この問題を単一話者発話区間検出（single-voice activity detection: SVAD）と定義し，これを実現する手法について検討する．本論文では特に，教師あり手法として深層ニューラルネットワーク（deep neural network: DNN）に基づく SVAD を提案する．提案手法は音響信号の時系列構造を最大限活用するために，DNN の一種である長・短期記憶ユニットを用いた再帰型ニューラルネットワークを SVAD に用いる．本研究では，日本人男女の音声データセットを用いて混合音声信号を作成し，この信号中の単一話者発話区間にラベル付けを行うことで教師データを作成する．様々な条件で実験を行った結果，提案手法はテストデータに対して 90% 程度の精度で単一話者発話区間が検出できることを確認した．

目次

第 1 章

緒言

1.1 本論文の背景

音源分離とは，観測信号をマイクロホンで測定し得られた混合音源から，混合前の信号を推定する技術である．具体的な応用例を Fig. 1.1 に示す．音源分離の例として音声信号に対する分離が挙げられる．音声信号に対する分離では，補聴器等に用いられている信号から雑音を除去して音声だけを抽出及び強調するタスクや，テレビの字幕等に用いられる複数話者の同時発話の分離などがある．しかしその一方で，周囲の（分離対象ではない）人の声，背景音楽，雑音などの影響で分離精度は急激に減少する [1]．音声認識技術を用いた製品が増えている中で，目的となる話者の音声以外の信号が混合することによる音声認識精度の低下を回避するためにも，音源分離を用いた目的話者の抽出・分離という前段処理が求められている．さらに音声認識だけでなく，イヤホンのノイズキャンセリング機能やテレビの字幕機能のように，人間の聴覚機能をサポートする面でも音源分離の応用先は数多く存在する．

前述のように，音源分離技術は重要な技術として長年研究されており，これらのタスクを満足するには高精度な音源分離手法が求められる．これまで発展してきた代表的な音源分離手法として，マイクロホンや音源の位置，音源数などの事前情報を用いずに，観測された混合音声信号のみから，混合前の音源信号を推定する手法であるブラインド音源分離（blind source separation: BSS）[2] がある．BSS の概要を Fig. 1.2 に示す．混合前の音源信号が，未知の混合系 $\boldsymbol{A}$ を経て混合しマイクロホンで観測される．BSS は，未知の混合系 $\boldsymbol{A}$ の逆系である分離系 $\boldsymbol{W}$ を観測信号のみから推定し，混合する前の各音源を得る技術である．

代表的な BSS の手法として，独立成分分析（independent component analysis: ICA）[3]，独立ベクトル分析（independent vector analysis: IVA）[4]，及び独立低ランク行列分析（independent low-rank matrix analysis: ILRMA）[5, 6] が存在する．ICA は，音源信号が時間領域で瞬時混合すること及び各音源信号が互いに統計的に独立であることの 2 点を仮定した BSS であり，混合信号を独立な成分（音源信号）に分離する．実際に観測した混合信号が，前述の 2 つの仮定を満たしている場合，ICA は高精度に BSS を達成できる．

しかしながら，実際の音響信号の混合は部屋の残響や各音源・各マイクロホン間の伝搬遅延の影響を受けて，時間領域では瞬時混合ではなく畳み込み混合となるため，ICA で音響信号の

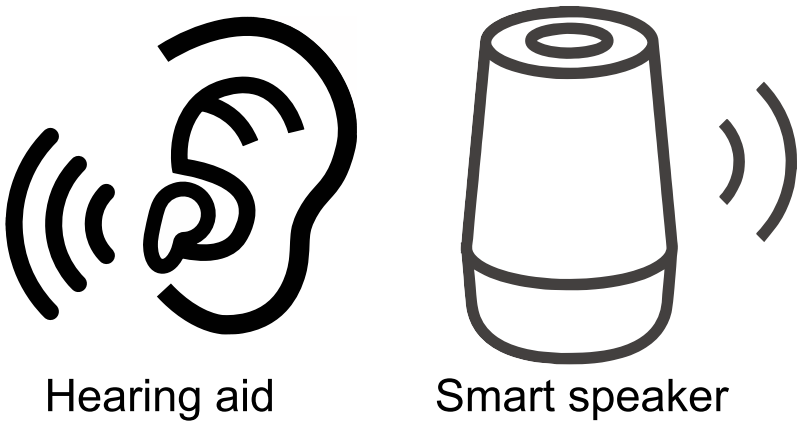


Fig. 1.1. Examples of application using speech source separation.

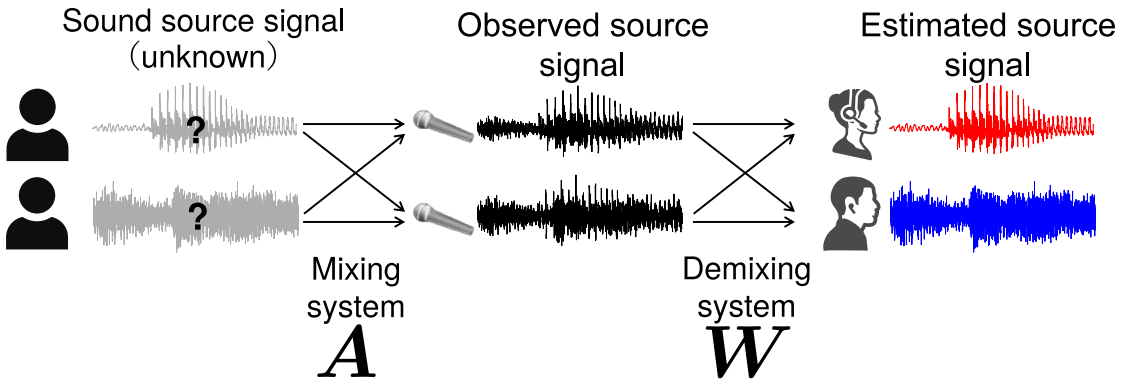


Fig. 1.2. Overview of BSS.

BSS を実現することはできない．この問題を解決した手法として，IVA や ILRMA 等が提案されており，これらは複数のマイクロホンを用いる BSS においてデファクトスタンダードなアルゴリズムとなっている．従って，IVA や ILRMA の音源分離性能を向上させることは重要であり，様々な改良・拡張が現在に至るまで提案されている [7, 8, 9, 10].

1.2　本論文の目的

IVA が高い分離性能を達成する条件を解析した論文として，文献 [11] がある．この文献では，複数の音源が混合した観測信号中に，特定の 1 音源のみが鳴っている時間区間が多く含まれているほど，IVA は高精度に分離系を推定できることが理論的に示されている．観測信号が複数話者の発話音声の混合信号である場合は，Fig. 1.3 に示すように，特定の 1 話者のみが発話している時間区間が多いほど，IVA は高い精度で複数話者を分離することができる．複数の話者が会話している状況は通常，発話者は話し手と聞き手の間で随時交代しながら会話が進むターンテイキングが行われるため，特定の 1 話者のみが発話している時間区間は多く含まれていると予想される．本論文では，この「特定の 1 話者のみが発話している時間区間」を「単一

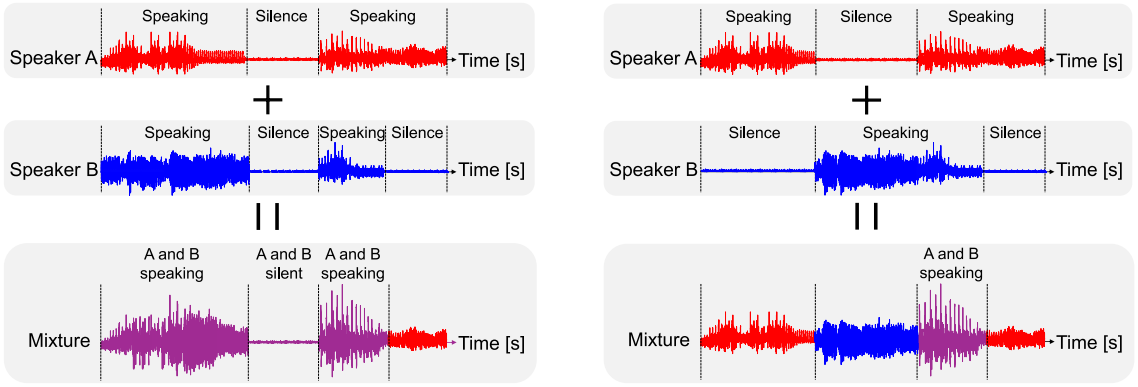


Fig. 1.3. Defference of mixture signals: observed mixture signal includes (a) many mixed segments and (b) many single-voice active segments. BSS performance of IVA is improved in case (b).

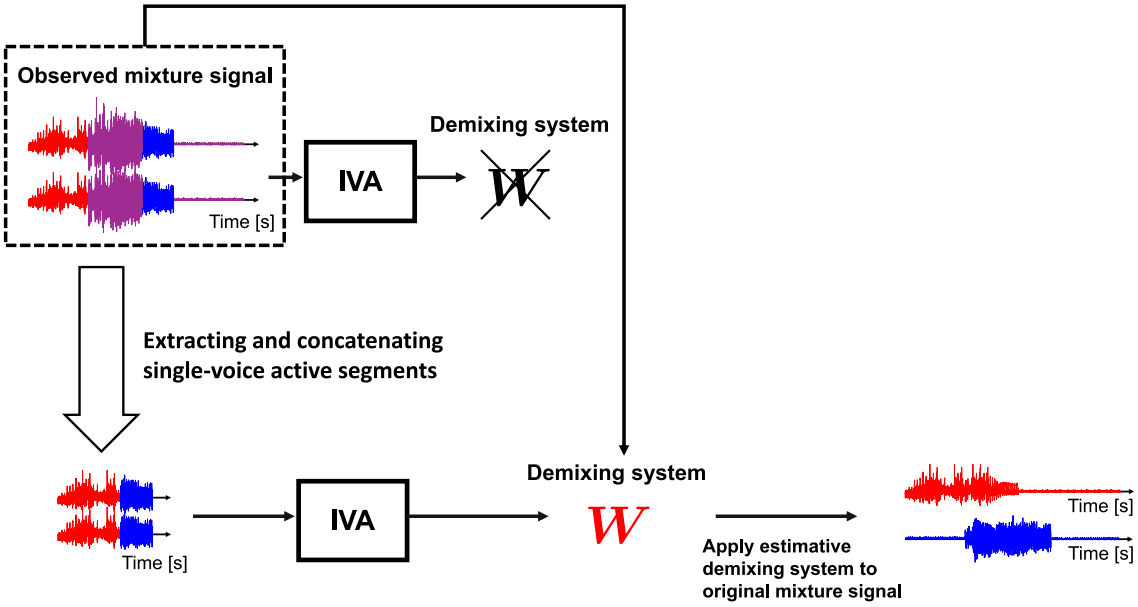


Fig. 1.4. More accurate separation utilizing only single-voice active segments.

話者発話区間」と定義する．

文献 [11] で解析された IVA の特徴を考慮すると，録音された音響信号の全時間区間を IVA の観測信号に与えるのではなく，Fig. 1.4 のように，複数の話者が同時に発話している時間区間を除去し単一話者発話区間のみを結合した音響信号を IVA の観測信号に与えた方が高精度な分離系が推定できると予想できる．推定された分離系は時間に対して非依存（時不変）であるため，録音された音響信号の全時間区間に対して適用することができ，全時間区間の高精度な分離信号が得られる．

前述のアルゴリズムを実現するためには，複数の話者が混合している音響信号から単一話者

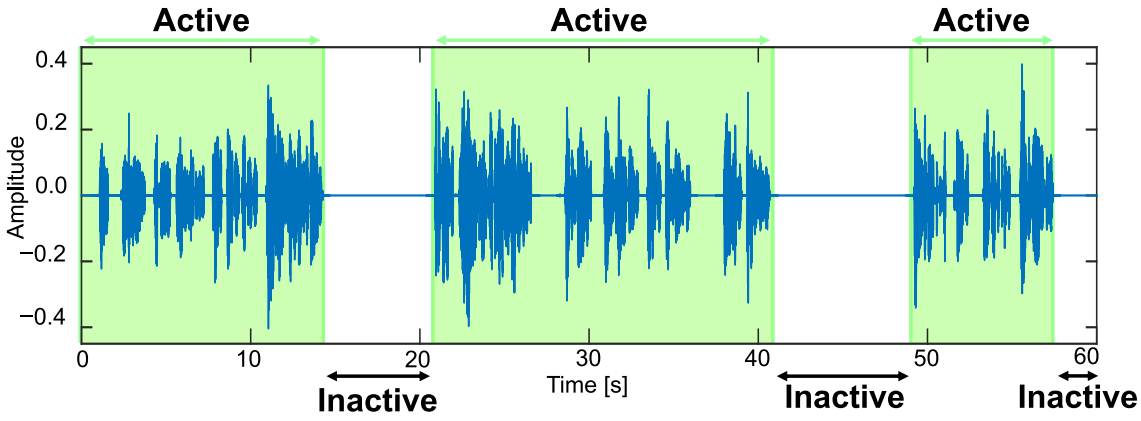


Fig. 1.5. Overview of voice activity detection.

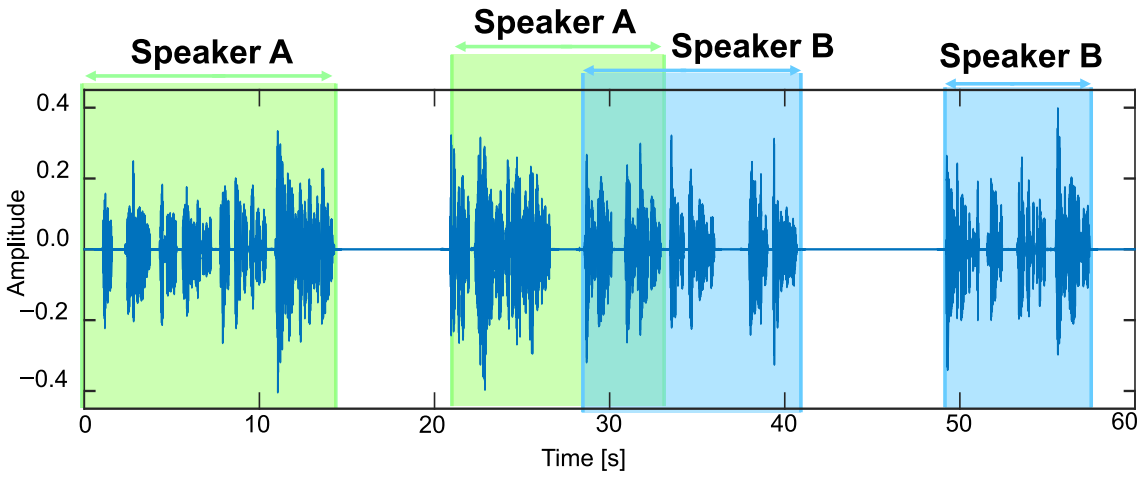


Fig. 1.6. Overview of speaker diarization.

発話区間を推定する必要がある．本論文では，この問題を「単一話者発話区間検出（single-voice activity detection: SVAD）」と定義し，SVAD を行う予測器の提案を目指す．具体的には，深層ニューラルネットワーク（deep neural network: DNN）に基づく SVAD について検討し，その学習方法や精度について議論する．

SVAD の類似技術として，Fig. 1.5 に示すように，観測信号から音声の発話されている時間区間のみを検出する音声区間検出（voice activity detection: VAD）[12] があり，音声通信における背景雑音抑圧や自動的なマイクロホンのオン/オフ等に利用される．本論文で取り扱う SVAD との違いは，SVAD は単一話者発話区間のみを検出することを目的としているのに対し，VAD は単一に限らず（複数話者であっても）音声の存在する時間区間を検出することを目的としている点にある．VAD では，単一話者発話区間の検出はできないため，前述の IVA の精度を向上するアルゴリズムへの活用はできない． その他，複数人が会話を行っている状況下で各話者の発話区間を検出しラベル付けする話者ダイアライゼーション（speaker diarization）[13] という技術もある．話者ダイアライゼーションを Fig. 1.6 に示す．この技

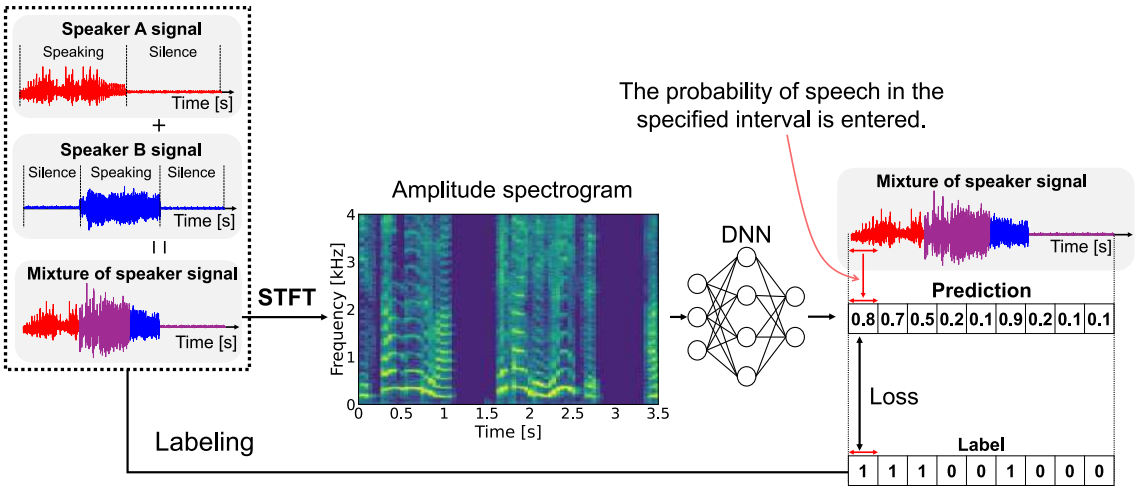


Fig. 1.7. Process flow of proposed method.

術は議事録作成や音源分離の前段処理に活用されるが，VAD や SVAD よりもさらに難易度の高い予測が求められるため，耐雑音性や検出精度の向上は大きな課題である．本論文では，VAD よりも複雑だが話者ダイアライゼーションよりは簡単な SVAD という問題に取り組むことで，比較的少ない量の学習データを用いて高精度に単一話者発話区間を検出できる手法の提案を目指す．提案手法の大まかな流れを Fig. 1.7 に示す．まず，データとして複数の話者の音声信号を混合し，観測信号とラベルを生成する．このとき，単一話者発話区間を 1，それ以外の区間を 0 とするラベル付けを行う．次に観測信号に短時間フーリエ変換（short-time Fourier transform: STFT）を適用して振幅スペクトログラムに変換し，DNN に入力する．DNN はできるだけラベルに近い結果を予測するように学習される．最終的に，学習には用いていないテストデータに対する予測精度について，実験を通して評価する．本論文では，観測信号の時間方向の構造を効率良く学習するために，長・短期記憶ユニットを用いた双方向再帰型ニューラルネットワーク（bidirctonal recurrent neural network using long-short term memory unit: BiLSTM）から成る DNN モデルを構築する．

1.3 本論文の構成

まず，2 章では，提案手法を理解するために重要な基礎知識である STFT 及び DNN について説明する．また，VAD や話者ダイアライゼーションの既存手法を類似研究として紹介し，提案手法との相違点について述べる．3 章では，本論文の提案手法について説明し，実験に用いる混合音声信号及び正解ラベルの作成方法を述べる．その後，本論文で用いる BiLSTM の構造及び学習方法について説明する．4 章では，実験として提案手法の DNN モデルを実際に学習し，単一話者発話区間の検出精度について評価と考察を行う．最後に 5 章では，本論文全体の結論を述べる．

第 2 章

基礎知識

2.1 まえがき

本章では，1 章で述べた提案手法を構成するために必要な基礎技術である，STFT，DNN，及び BiLSTM をそれぞれ 2.2 節，2.3 節，及び 2.4 節で詳細に述べる．2.5 節では本論文で取り扱う SVAD という問題に類似する研究について述べる．2.6 節で本章をまとめる．

2.2 STFT

STFT は，音響信号の時間的に変化するスペクトルを，時間周波数領域と呼ばれる二次元の特徴量空間で表現するための変換手法である．STFT の概要を Fig. 2.1 に示す．STFT では，音響信号の時間波形を短時間区間に分割し，窓関数を乗じたうえで周波数領域へと変換する．音響信号の時間波形を次式で定義する．

$$\boldsymbol{y} = [y(1), y(2), \cdots, y(l), \cdots, y(L)]^{\mathrm{T}} \in \mathbb{R}^{L} \tag{2.1}$$

ここで，$\cdot^{\mathrm{T}}$ は転置，L は時間信号 $\boldsymbol{y}$ の長さ，$l = 1, 2, \cdots, L$ は時間信号 $\boldsymbol{y}$ の離散時間サンプルをそれぞれ表す．短時間区間長（窓長）及び短時間区間のシフト長をそれぞれ Q 及び τ としたとき，時間領域の信号時間領域の信号 $\boldsymbol{y}$ の j 番目の短時間区間（時間フレーム）の信号 $\tilde{\boldsymbol{y}}^{(j)}$ は次式で表される．

$$\begin{aligned} \tilde{\boldsymbol{y}}^{(j)} &= [y((j-1)\tau + 1), y((j-1)\tau + 2), \cdots, y((j-1)\tau + Q)]^{\mathrm{T}} && (2.2) \\ &= [\tilde{y}^{(j)}(1), \tilde{y}^{(j)}(2), \cdots, \tilde{y}^{(j)}(q), \cdots, \tilde{y}^{(j)}(Q)]^{\mathrm{T}} \in \mathbb{R}^{Q} && (2.3) \end{aligned}$$

ここで，$j = 1, 2, \cdots, J$ 及び $q = 1, 2, \cdots, Q$ は，それぞれ時間フレーム及び時間フレーム内のサンプルを示す．また，フレーム数 J は次式によって与えられる．

$$J = \frac{L}{\tau} \tag{2.4}$$

ただし，信号長 L はフレーム数 J が整数となるように各時間フレームの信号の両端にゼロを挿入する処理 （ゼロパディング） が施されている．このとき時間フレームの信号を全ての j

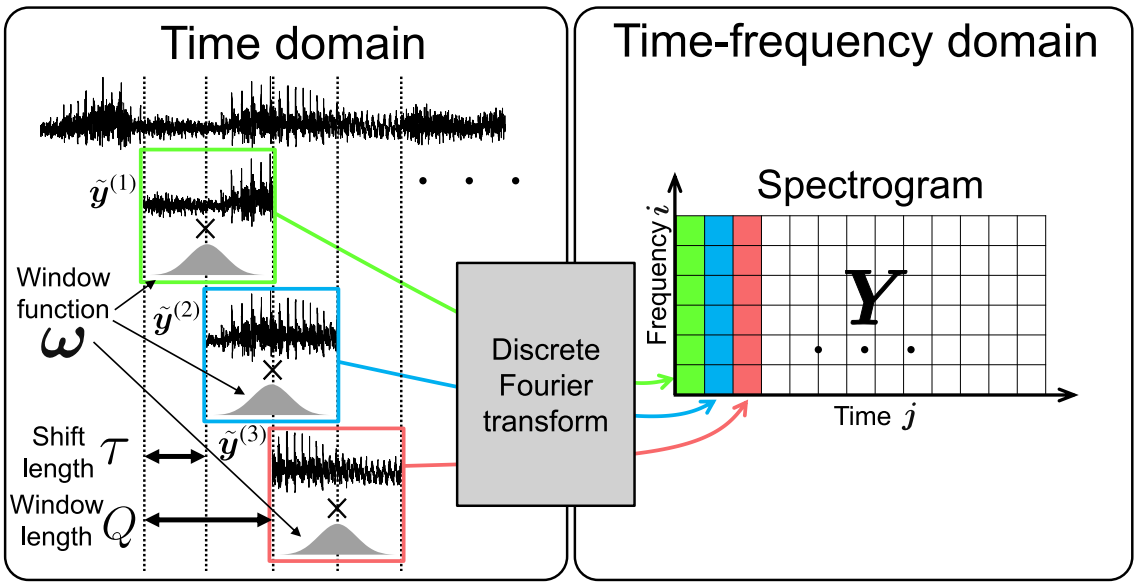


Fig. 2.1. Mechanism of STFT.

についてまとめた全時間フレームの信号は次式の通り定義できる．

$$\tilde{\boldsymbol{Y}} = [\tilde{\boldsymbol{y}}^{(1)}\ \tilde{\boldsymbol{y}}^{(2)}\ \cdots\ \tilde{\boldsymbol{y}}^{(j)}\ \cdots\ \tilde{\boldsymbol{y}}^{(J)}] \in \mathbb{R}^{Q\times J} \tag{2.5}$$

次に，長さ Q の窓関数を $\boldsymbol{\omega} = [\omega(1), \omega(2), \cdots, \omega(q), \cdots, \omega(Q)]^{\mathrm{T}} \in \mathbb{R}^{Q}$ と定義する．STFT の処理は次式で表される．

$$\boldsymbol{Y} = \mathrm{STFT}_{\boldsymbol{\omega}}(\tilde{\boldsymbol{Y}}) \in \mathbb{C}^{I\times J} \tag{2.6}$$

$$y_{ij} = \sum_{q=1}^{Q} \omega(q) y^{(j)}(q) \exp\left\{\frac{-\iota 2\pi (q-1)(i-1)}{Q}\right\} \tag{2.7}$$

ここで，$\boldsymbol{Y}$ は複素スペクトログラムと呼ばれ，複素数の時間周波数成分を持つ行列である．また，y_{ij} は $\boldsymbol{Y}$ の (i,j) 要素を表す．I は $I = \lfloor \frac{Q}{2} \rfloor + 1$ を満たす整数（$\lfloor\cdot\rfloor$ は床関数），$i = 1, 2, \cdots, I$ は周波数ビンのインデクス，$j = 1, 2, \cdots, J$ は時間フレームのインデクス，ι は虚数単位を示している．このように，時間領域の信号を一定幅 Q の短時間毎に区切って分析窓関数 $\boldsymbol{\omega}$ を乗じて離散フーリエ変換（discrete Fourier transformation: DFT）することで，周波数と時間の 2 次元複素行列であるスペクトログラム $\boldsymbol{Y}$ に変換できる．複素スペクトログラムは各時間周波数の振幅成分と位相成分を持っているが，音源分離等の多くの音響信号処理では，振幅成分のみを取り扱うことが多い．その場合は，複素スペクトログラム $\boldsymbol{Y}$ の各要素に関して絶対値を取った振幅スペクトログラム $|\boldsymbol{Y}| \in \mathbb{R}_{\geq 0}^{I\times J}$ や，絶対値の 2 乗を取ったパワースペクトログラム $|\boldsymbol{Y}|^{\cdot 2} \in \mathbb{R}_{\geq 0}^{I\times J}$ を処理の対象とする．ここで，ベクトルや行列に対する絶対値記号及びドット付き指数乗はそれぞれ要素毎の絶対値及び要素毎の指数乗を表す．本論文では振幅スペクトログラムを用いる．

Fig. 2.2 に音声波形を STFT し得られた振幅スペクトログラムを示す．但し，振幅値は対数を取ってデシベルに変換し，色の違いで表現している．音声信号は一般に，基本周波数成分と

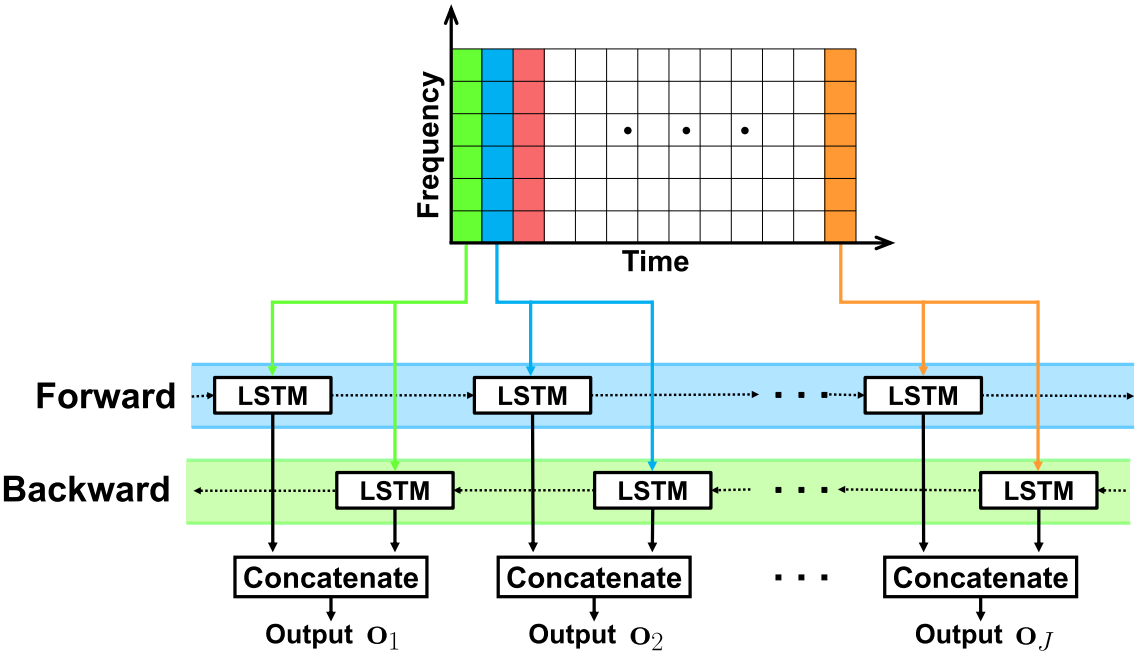


Fig. 2.2. Example of amplitude spectrogram of speech signal.

その整数倍の高次成分が同時に生起するため，Fig. 2.2 のように大きな振幅を持つ成分が縞模様となって現れる．

2.3　DNN

DNN は，脳の神経回路を模したニューラルネットワークをより深い階層に適応させたものである．DNN を用いた機械学習は深層学習とよばれ，画像認識や機械翻訳等に利用されている．Fig. 2.3 には単純パーセプトロンを表す．単純パーセプトロンは入力層と出力層のみから構成されるシンプルなネットワークであり，各入力に重みをつけて合計した値に非線形関数を通して出力を得る．

いま，単純パーセプトロンの入力ベクトルを $\mathbf{x} = [\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_\mathrm{n}, \cdots \mathrm{x}_\mathrm{N}]^\mathrm{T} \in \mathbb{R}^\mathrm{N}$ とおく．単純パーセプトロンの出力 y は，重み係数ベクトル $\mathbf{w} = [\mathrm{w}_1, \mathrm{w}_2, \cdots, \mathrm{w}_\mathrm{n}, \cdots, \mathrm{w}_\mathrm{N}]^\mathrm{T} \in \mathbb{R}^\mathrm{N}$，バイアス $\mathbf{b} = [\mathrm{b}_1, \mathrm{b}_2, \cdots, \mathrm{b}_\mathrm{n}, \cdots, \mathrm{b}_\mathrm{N}]^\mathrm{T} \in \mathbb{R}^\mathrm{N}$ を用いて次式で表される．

$$\mathrm{y} = \begin{cases} 0 & (\text{if } \mathbf{w}^\mathrm{T}\mathbf{x} + \mathbf{b} \geq 0) \\ 1 & (\text{otherwise}) \end{cases} \tag{2.8}$$

この単純パーセプトロンはニューラルネットワークの非常に基礎的な構成部品である．DNN で用いられるニューロンモデルは，この単純パーセプトロンを一般化した形となっている．ニューロンモデルの出力は次式で与えられる．

$$\mathrm{y} = \phi(\mathbf{w}^\mathrm{T}\mathbf{x} + \mathbf{b}) \tag{2.9}$$

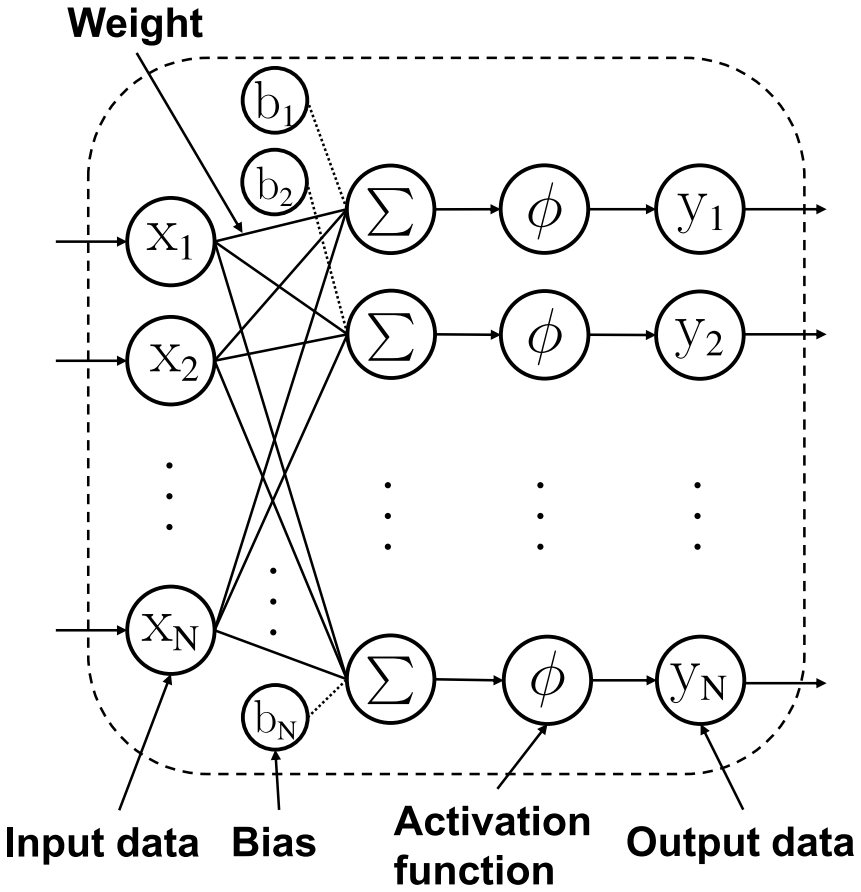


Fig. 2.3. Simple and multi-layer perceptrons.

ここで，$\phi(\cdot)$ は活性化関数と呼ばれる何らかの非線形関数である．このニューロンモデルを Fig. 2.3 のように多重に結合し，さらにこれを Fig. 2.4 のように多層に結合したネットワークが多層パーセプトロン（multi-layer perceptron: MLP）と呼ばれる DNN の基本形となる．学習データを用いてネットワーク中のすべての重み係数を適切に調整することで，望ましい結果が得られる分類器や予測器を構築することができる．また，DNN 中に使われる活性化関数の一例を Fig. 2.5 に示す．いずれも非線形な関数であるが，DNN の学習時の勾配消失問題に対処するために ReLU やそれによく似た活性化関数が比較的よく用いられる．

DNN の学習とは，入力に対する DNN の出力結果を正解と合致するように各ニューロンの重みを調整することをいう．Fig. 2.6 に DNN の学習の概要を示す．Fig. 2.6 のようにベクトルを入力した場合に出力データは $[0.7, 0.3]^{\mathrm{T}}$ が出力されている．これに対して，予測の正解となるラベルは $[1, 0]^{\mathrm{T}}$ であるので，出力データ（予測結果）が正解データ（ラベル）に近づくようにニューロンの重みとバイアス値の調整を行う．学習が進むと，$[0.9, 0.1]^{\mathrm{T}}$ となり正解データに近い値となる．この図では 1 つの入力データとそれに対応するラベルのみが描かれているが，実際の学習においては，膨大な入力データとそのラベルを用いて常に正しい予測が得られるように学習する．また，その結果，学習時には入力されなかったデータ（テストデータ）に

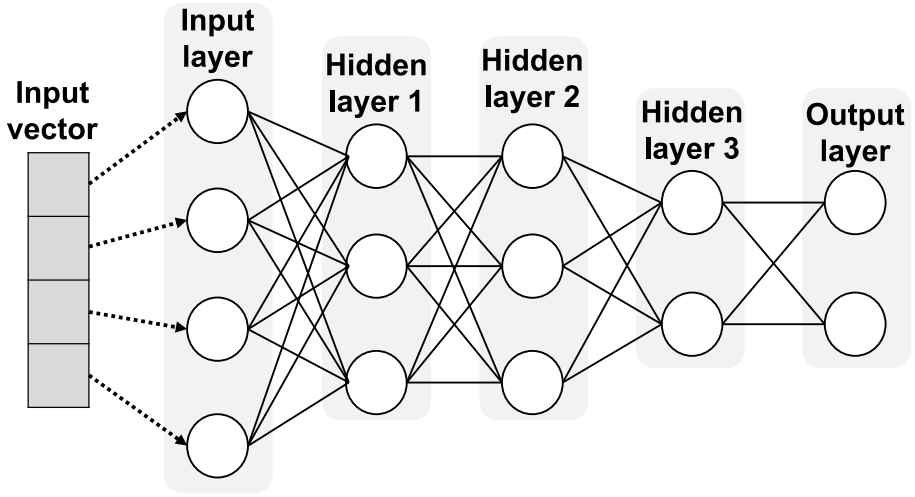


Fig. 2.4. Architecture of MLP.

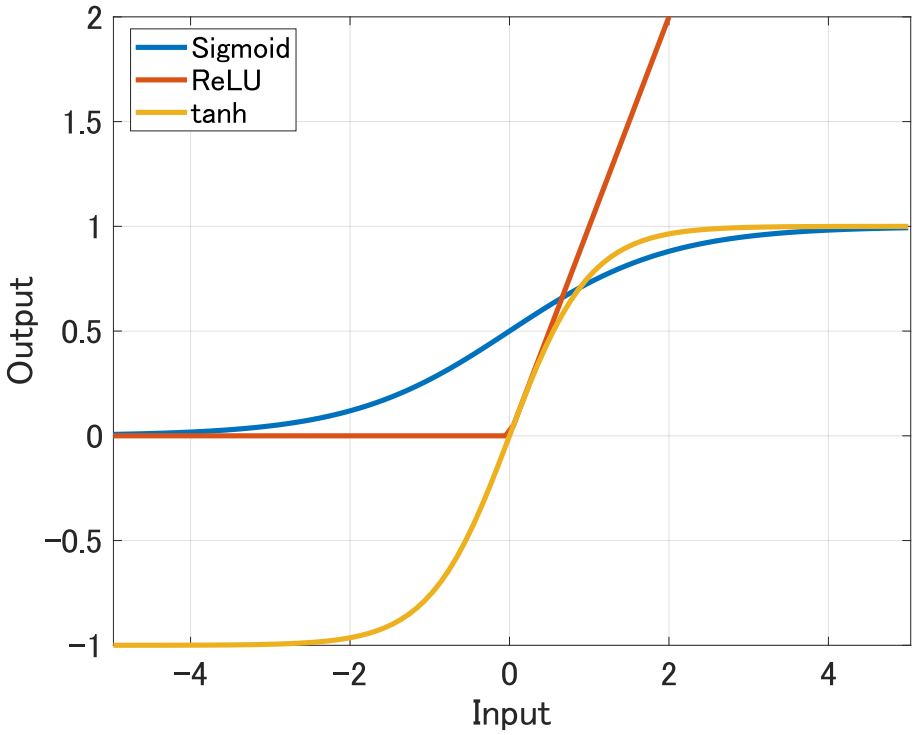


Fig. 2.5. Examples of activation function.

対しても望ましい出力を正確予測できるようになることが求められる．このような未知のデータに対する予測性能は汎化性能と呼ばれる．また，Fig. 2.6 に記載されている損失は機械学習モデルが算出した予測値と実際の正解値のズレを計算する関数である．従って，DNN の精度を高める上で最終的なゴールは，損失の値を学習データとテストデータの両方に対して可能な限り小さくしていくことである．学習時には，誤差逆伝播と呼ばれる損失の勾配値の伝搬を行い，損失を小さくする方向に全ての層の重みを調整する．

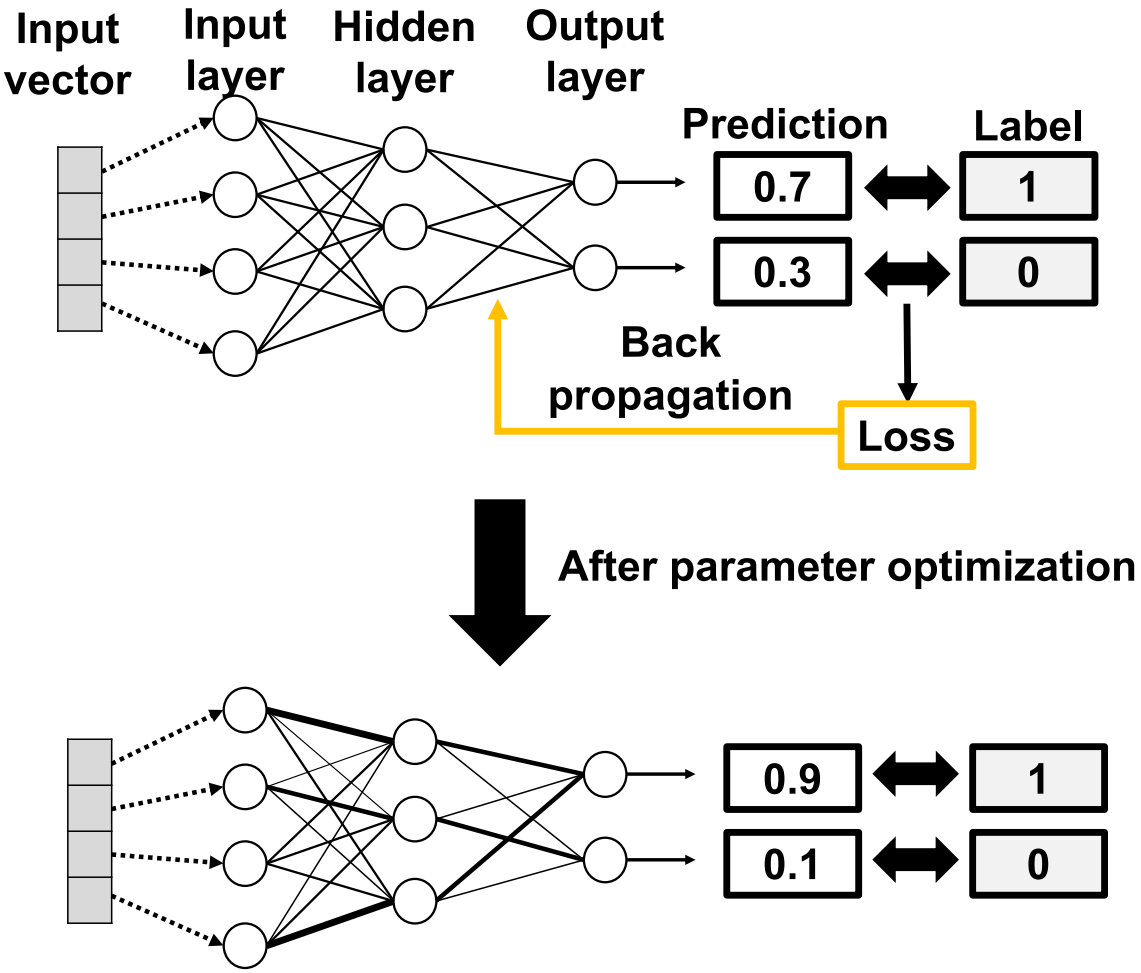


Fig. 2.6. Training of DNN.

2.4 BiLSTM

時系列のデータを扱う DNN として，再帰型ニューラルネットワーク（recurrent neural network: RNN）がある．これは，Fig. 2.7 のように，ある時刻のネットワークの出力を次の時刻の入力に加える構造を持った DNN である．図中の $\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}$ はそれぞれ時刻 $t-1$，t，$t+1$ の入力データベクトル，$\mathbf{h}_{t-1}, \mathbf{h}_t, \mathbf{h}_{t+1}$ はそれぞれ時刻 $t-1$，t，$t+1$ の内部状態，$\mathbf{y}_{t-1}, \mathbf{y}_t, \mathbf{y}_{t+1}$ はそれぞれ時刻 $t-1$，t，$t+1$ の出力ベクトルを表す．このような構造を持つ DNN を考えることで，時系列方向の構造を効率的に学習することができる利点がある．通常の RNN は過去から未来の一方向の構造を持つが，この RNN を Fig. 2.8 のように順方向と逆方向で 2 つ組み合わせることで，未来から過去の方向も含めて双方向の学習が可能となる．このモデルを双方向 RNN（bidirectional RNN: BiRNN）と呼ぶ．

次に，RNN においてよく用いられるアーキテクチャの長・短期記憶（long short term memory: LSTM）ユニット [14] について説明する．LSTM は，RNN の一種であり，通常の

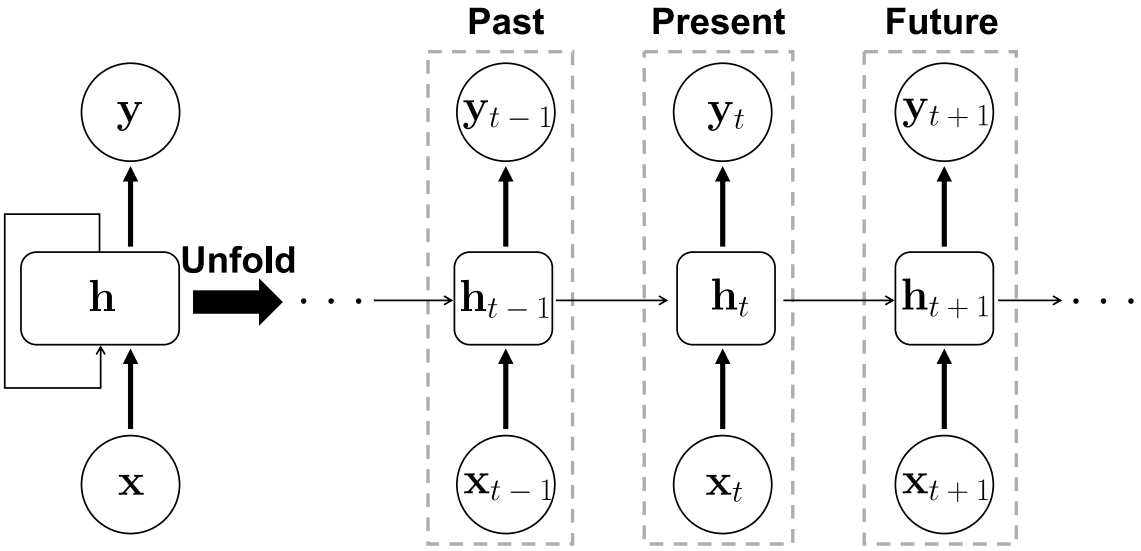


Fig. 2.7. Architecture of RNN.

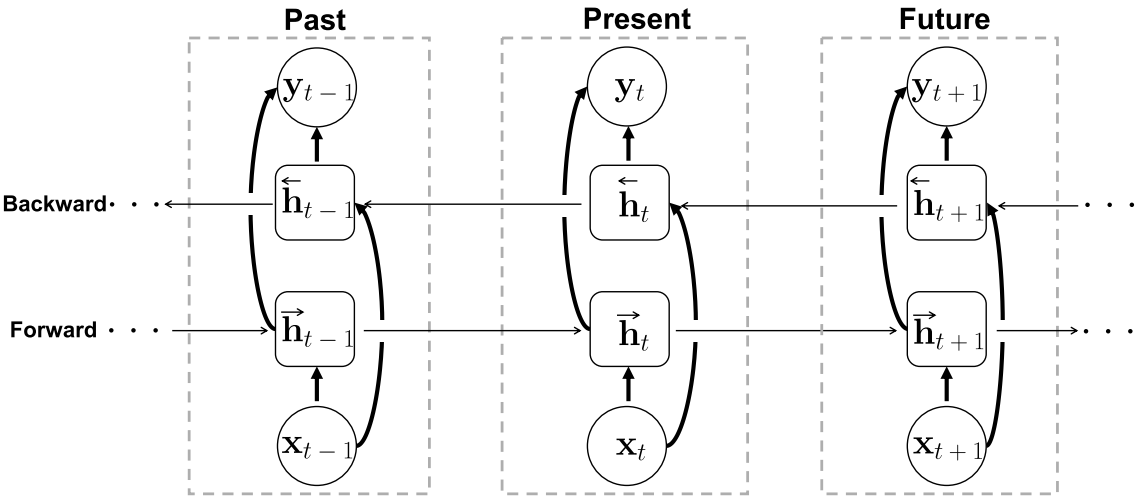


Fig. 2.8. Architecture of BiRNN.

RNN では難しい時系列データの長期的な依存関係を学習することができる構造を持っている．LSTM の内部構造を Fig. 2.9 に示す．LSTM は「ゲート」という概念を導入し，情報が長期間にわたって保持されるか，忘れられるかを決定する能力を持っている．これにより，LSTM は長い時間スケールでの依存関係を捉え，時系列データの複雑な構造やパターンを効率的に学習することが可能となる．時刻 t における，LSTM ユニットへの入力データベクトルを $\mathbf{x}_t$ 及び $\mathbf{h}_t$ とすると，LSTM ユニットでの計算は次のようになる．

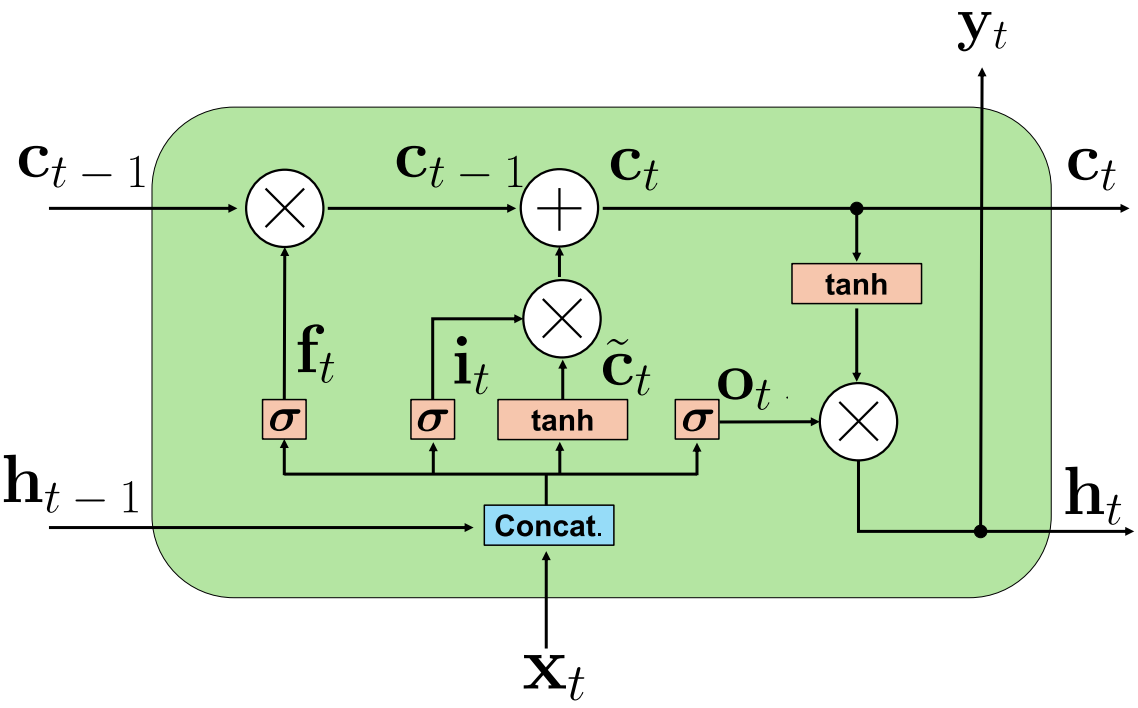


Fig. 2.9. Structure of LSTM unit.

$$\mathbf{f}_t = \sigma(\mathbf{W}^{(\mathbf{f})}\mathbf{x}_t + \mathbf{R}^{(\mathbf{f})}\mathbf{h}_{t-1} + \mathbf{b}^{(\mathbf{f})}) \tag{2.10}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}^{(\mathbf{i})}\mathbf{x}_t + \mathbf{R}^{(\mathbf{i})}\mathbf{h}_{t-1} + \mathbf{b}^{(\mathbf{i})}) \tag{2.11}$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}^{(\tilde{\mathbf{c}})}\mathbf{x}_t + \mathbf{R}^{(\tilde{\mathbf{c}})}\mathbf{h}_{t-1} + \mathbf{b}^{(\tilde{\mathbf{c}})}) \tag{2.12}$$

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t \tag{2.13}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}^{(\mathbf{o})}\mathbf{x}_t + \mathbf{R}^{(\mathbf{o})}\mathbf{h}_{t-1} + \mathbf{b}^{(\mathbf{o})}) \tag{2.14}$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}^{(t)}) \tag{2.15}$$

ここで，$\sigma(\cdot)$ はベクトルの各要素に対するシグモイド関数，$\mathbf{W}^{(\mathbf{f})}$，$\mathbf{W}^{(\mathbf{i})}$，$\mathbf{W}^{\tilde{\mathbf{c}}}$，及び $\mathbf{W}^{(\mathbf{o})}$ は時間 t における入力ベクトル $\mathbf{x}_t$ に対する重み係数行列，$\mathbf{R}^{(\mathbf{f})}$，$\mathbf{R}^{(\mathbf{i})}$，$\mathbf{R}^{\tilde{\mathbf{c}}}$，及び $\mathbf{R}^{(\mathbf{o})}$ は，時間 $t-1$ における出力ベクトル $\mathbf{h}_{t-1}$ に対する重み係数行列，$\mathbf{b}^{(\mathbf{f})}$，$\mathbf{b}^{(\mathbf{i})}$，$\mathbf{b}^{\tilde{\mathbf{c}}}$，及び $\mathbf{b}^{(\mathbf{o})}$ は，それぞれの係数に対するバイアスベクトル，$\tanh(\cdot)$ はベクトルの各要素に対する双曲線正接関数，$\circ$ はベクトルの要素毎の積をそれぞれ示す．$\mathbf{f}_t$ は，忘却ゲートと呼ばれ，時間 $t-1$ における長期記憶ベクトル $\mathbf{c}_{t-1}$ からどの要素を保持するか決定するベクトルである．$\mathbf{i}_t$ は入力ゲートと呼ばれ，時間 t における入力ベクトル $\mathbf{x}_t$ 及び時間 $t-1$ における出力ベクトル $\mathbf{h}_{t-1}$ からどの要素を保持するか決定するベクトルである．$\mathbf{o}_t$ は出力ゲートと呼ばれ，時間 t における出力ベクトル $\mathbf{h}_t$ を求めるためのベクトルである．忘却ゲート $\mathbf{f}_t$ 及び入力ゲート $\mathbf{i}_t$ を用いて，時間 t における長期記憶 $\mathbf{c}_t$ が得られる．これを順方向及び逆方向で行うことで時系列データとして学習する．上記は順方向の場合であり，逆方向の場合，時間 t における長期記憶ベクトル $\mathbf{c}_t$ 及び短期記憶ベクトル $\mathbf{h}_t$ は，時間 $t+1$ における長期記憶ベクトル $\mathbf{c}_{t+1}$ 及び短期記憶ベクトル $\mathbf{h}_{t+1}$ を用いて求める．

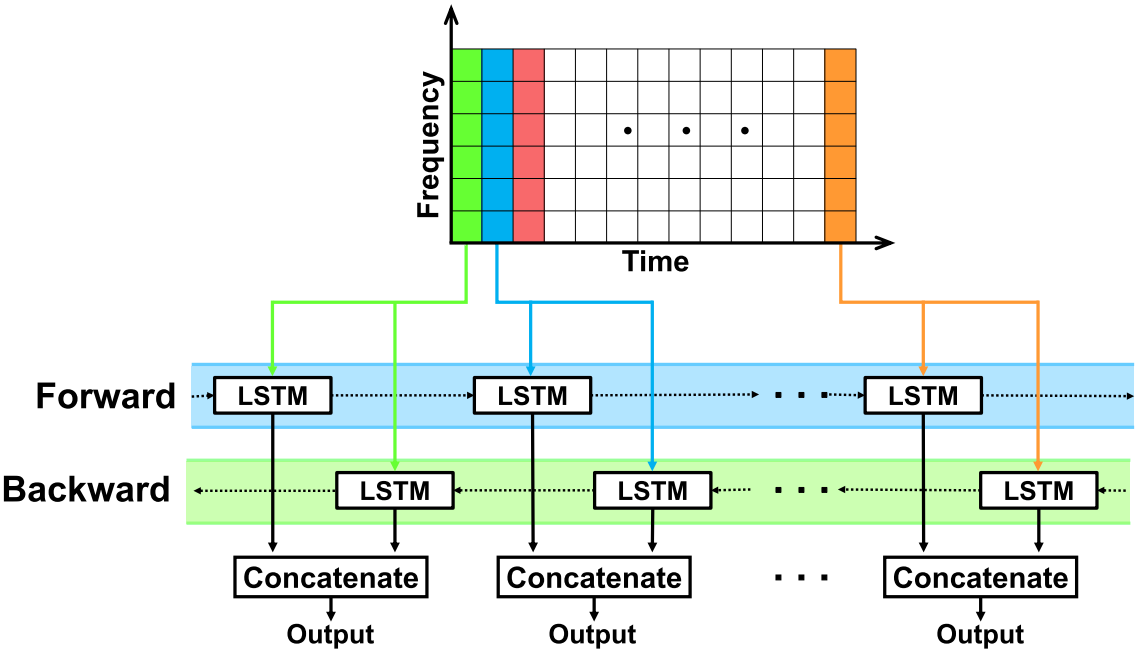


Fig. 2.10. Architecture of BiLSTM.

Fig. 2.10 に BiLSTM の構造を示す．LSTM ユニットに基づく RNN を順方向処理の Forward と逆方向処理の Backward にそれぞれ適用し，両方向の出力から最終的な出力を予測する．場合によっては，両方向の出力をさらに MLP につなげて出力を得る場合もある．

2.5　既存の類似研究

本節では, 本論文で取り扱う SVAD という問題に類似する研究について説明する．VAD [15] では音声の発話区間と非発話区間の検出を行う．話者ダイアライゼーション [16] では話者情報とその話者に対する発話区間の推定を行う．本論文で解決を目指す SVAD は単一話者発話区間の推定が必要なので，VAD よりは複雑だが，話者ダイアライゼーションよりは単純な問題である．話者ダイアライゼーションを用いても，1 章で述べた BSS の性能向上に応用できると考えられるが，この応用は話者ダイアライゼーションほどの高度な推定は必要としていない．そこで本論文は，必要最小限の問題である SVAD の解決を目指すことで，より少ない学習コストで目的を達成することを目指している．話者ダイアライゼーションのために提案されている手法と提案法の性能比較については，今後の課題とする．

2.5.1　VAD

VAD [15] とは，音声や雑音等の信号が含まれる観測信号の中から，音声信号の含まれる時間区間（音声区間）とそれ以外の時間区間（非音声区間）を判別する技術である [12, 17]VAD は DNN を使わない古典的な手法 [18] と DNN を使う手法 [19] が存在する．文献 [18] では，

最も初期に提案された観測信号のパワーと零交差数を用いる手法が提案されている．この文献では有声音を検出するために信号のパワー，無声子音等を取り出すために零交差数を用い，事前に設定した閾値に基づいてこれらを処理することで VAD を行っている．文献 [19] では，DNN を用いた VAD モデルの手法を提案されている．この文献ではゲート付き回帰型ユニットを用いた RNN，時間的畳み込みネットワーク，及びトランスフォーマーエンコーダベースのネットワークの学習を行い VAD の検出を行っている．提案手法との関連性として深層学習を用いて音声区間検出を行うことである．本実験では時間方向の構造を効率的に学習することを目的としているため，BiLSTM を用いる．

本論文が対象にしている SVAD と VAD の相違点として，SVAD では単一話者発話区間の推定を行うことが目的であることに対して，VAD は信号内の波形を発話区間と非発話区間にのみ分類することを目的としている．VAD は話者ごとの発話区間推定を行っていないため，複数の話者が同時に発話している時間区間も発話区間として推定される技術である．1.2 節で述べた BSS への応用を考えると，事前処理として VAD ではなく SVAD が必要とされるため，本節で紹介した手法は基本的に本研究の目的達成には利用することができない．

2.5.2 話者ダイアライゼーション

話者ダイアライゼーション [20, 21, 22, 23] は，音声の長さや話者数等の条件が不明の状態から，いつ，誰が話しているかを推定する技術である．VAD との違いとして，まず VAD は「いつ話したか」はわかるが，「誰が話したか」の情報がない．話者ダイアライゼーションは，発話区間判別に加え，話者情報も推定する．話者ラベルを使用して書き起こしを強化する音声認識システムにとって不可欠な機能である．「1 誰がいつ話したか」を把握するために，話者ダイアライゼーションシステムは，目に見えない話者の特徴を捕捉でき，音声録音のどの領域がどの話者に属しているかを区別する必要がある．これを実現するために，話者ダイアライゼーションシステムでは各人の音声の特徴量を抽出し，話者の数を数え，音声セグメントを対応する話者インデクス（話者ラベル）に割り当てている．話者ダイアライゼーションには DNN を使わない古典的な方法 [24] と DNN を使う方法 [20, 25] が存在する．文献 [24] では，マルチステージのセグメンテ―ジョン及びクラスタリングシステムを用いた話者ダイアライゼーションの提案をしている．反復ガウス混合モデル（gaussian mixture model: GMM）クラスタリングを，ベイジアン情報量基準（bayesian information criterion: BIC）クラスタリングに置き換え，GMM ベースのスピーカー識別メソッドを使用したクラスタリングステージが Viterbi アルゴリズムに追加された．文献 [25] では，可変長の音声セグメントに対して固定次元の埋め込みを学習し，セグメントが同じ話者または異なる話者から派生した可能性を測定するスコアリング関数も同時に学習する．DNN を用いる研究は他にもある．文献 [20] では，会話の中には話者のターンが非常に短い相槌などが原因で，通常のダイアライゼーションシステムでは機械での文字起こしが非常に困難であることがあげられている．解消するために，オーディオをセグメント化する一方で，話者ダイアライゼーションでは数十分の 1 秒から数秒の比較的短い

セグメントについてきめ細かい決定を行う必要がある．だが，非常に短いセグメントから信頼できる話者の特徴を捕捉する可能性が低いため，短い音声セグメントに対して正確かつきめ細かい決定を下すことは困難である．その方法として，ダイアライゼーションのマルチスケールモデルが提案された．VAD 用の MarbleNet モデル [26]（音声コマンド認識用の end-to-end neural network）と話者埋め込み抽出用の TitaNet モデル [27]（話者表現を抽出するための ContextNet アーキテクチャ）と呼ばれる深層学習を用いた手法となっている．

話者ダイアライゼーションでは，理想的には各時間区間にどの話者が発話しているかが得られるため，1.2 節で述べた BSS への応用にも活用することが可能である．しかしながら，話者ダイアライゼーションそのものが比較的難しい問題であるため，高精度な話者ダイアライゼーションを達成するためには比較的大きい規模の DNN モデルや大量の学習データが必要となるリスクがある．また，1.2 節で述べた BSS への応用にはすべての話者の発話区間情報を必要としておらず，単一話者発話区間さえ事前推定できれば良い．従って本論文では，話者ダイアライゼーションよりも簡単な問題である SVAD の解決のみを目指す．解くべき問題が話者ダイアライゼーションよりも簡単になることから，必要となる学習データの量や DNN の複雑性を抑えることができるため，より簡便な SVAD の解決ができることを期待している．

2.6　本章のまとめ

本章では，1 章で説明した提案手法を理解するために必要となる基礎知識及び提案手法の類似研究について説明した．2.2 節では STFT について，2.3 節では DNN について説明した．2.4 節では実験で使用する BiLSTM について説明した．2.5 節では VAD や話者ダイアライゼーションの既存手法を類似研究として紹介し，提案手法との相違点について述べた．3 章では，提案手法の詳細と実装方法について説明し，ネットワークモデルを作成する．

第 3 章

提案手法

3.1 まえがき

本章では，1 章で述べた SVAD を実現する手法として，BiLSTM を用いた手法を 3.2 節で提案し，その具体的な単一話者発話区間の推定方法について説明する．また，提案手法を学習するための入力信号（混合音声信号）とそのラベル（単一話者発話区間情報）の作成方法について 3.3 節で述べる．さらに，具体的なネットワークモデルの構造を 3.4 節で述べ，その学習方法を 3.5 節に記載する．3.6 節で本章をまとめる．

3.2 DNN に基づく SVAD

本節では，本論文の提案手法の概要を説明する．SVAD は観測された複数話者の混合音声信号から単一話者発話区間を推定することが目的である．本論文で扱うデータは時系列データであるため，2.4 節で説明した BiLSTM を用いることで時間方向の構造を効率的に学習することを目指す．

提案手法の概要を Fig. 3.1 に示す．この図の上から順に詳細を説明する．まず混合音声信号の入力を行う．本実験では 2 人の話者の発話を含む混合音声信号を入力と想定している．また，入力の混合音声信号には予測の正解となるラベルが付与されている．ラベルとして与えられている情報は，混合音声信号を振幅スペクトログラムに変換した際の各時間フレームが単一話者発話区間か否かという 2 値の情報である．入力の混合音声信号とそれに対応するラベルの作成方法については，次節で詳細を述べる．

まず，混合音声信号に式 (2.6) の STFT を適用し，振幅スペクトログラム $|\boldsymbol{Y}|$ を求める．この振幅スペクトログラム $|\boldsymbol{Y}|$ を J 個の時系列の I 次元ベクトルとみなし，Fig. 3.1 のように BiLSTM に入力する．BiLSTM 内部の次元数や層数等については 3.4 節で詳しく説明する．BiLSTM では時系列の入力信号に対して，順方向と逆方向の双方向の処理が行われる．BiLSTM の順方向と逆方向の出力ベクトルは 1 つのベクトルとして結合（concatinate）され，さらに全結合層（dense layer）に入力される．この全結合層に入力されたベクトルは，全結合

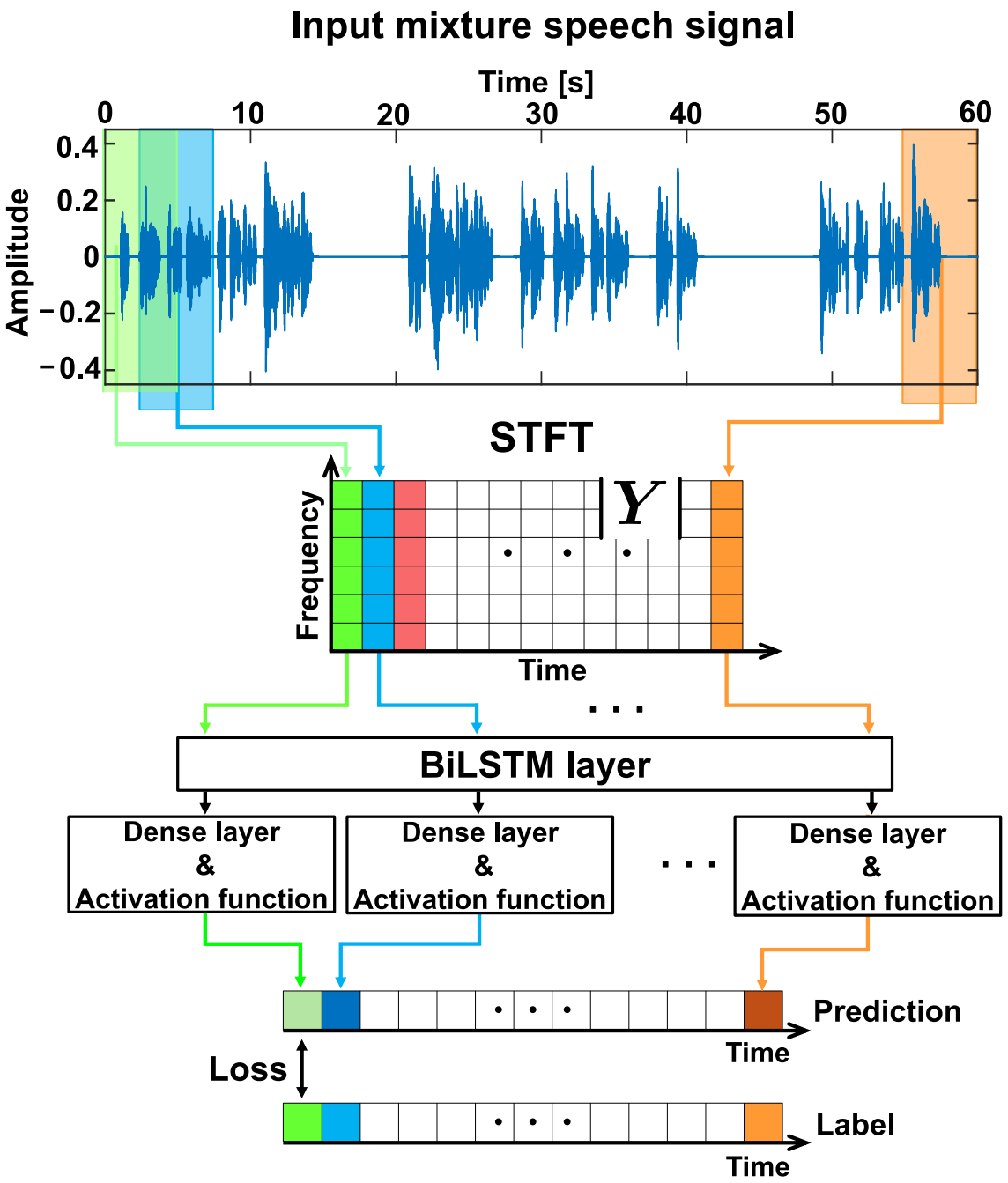


Fig. 3.1. Process flow of proposed method.

層を 1 層通し，最終的に softmax の活性化関数を経由することで単一話者発話区間である確率と単一話者発話区間ではない確率の 2 値に変換されて，予測結果として出力される．この予測結果とラベルが一致するように BiLSTM 及び全結合層中の全てのパラメータが学習・更新される．

3.3 教師データと混合音声信号の作成

本節では，前節で述べたネットワーク全体を学習するうえで必要な教師データである混合音声信号及び正解ラベルの作成方法について説明を行う．本論文では，日本語多数話者音声コーパスである Japanese versatile speech corpus (JVS corpus) [28] を使用し，混合音声信号と正解ラベルを作成する．

まず，コーパスに収録されている一人の話者の N 種類の発話音声信号を用いて，単一話者の観測信号を作成する．Fig. 3.2 に単一話者の観測信号の作成方法を示す．本研究で作成する観測信号を $\boldsymbol{a} \in \mathbb{R}^L$ とし，その長さを L とする．この観測信号 $\boldsymbol{a}$ の作成に使用する同一話者の N 個の発話音声信号を $(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \cdots, \boldsymbol{u}^{(n)}, \cdots, \boldsymbol{u}^{(N)})$ と定義する．この発話音声信号の長さは異なるが，全て L よりは短いものとする．この発話音声信号 $\boldsymbol{u}^{(n)}$ の時間 l における信号値を $u^{(n)}(l)$ とおく．N 個の発話音声信号 $(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \cdots, \boldsymbol{u}^{(n)}, \cdots, \boldsymbol{u}^{(N)})$ が，Fig. 3.2 のように，$\boldsymbol{a}$ の時間インデクスにおいてお互いに重なりあわないように時間遅れを与えたうえで，足し合わせた信号を，ある話者の観測音声信号 $\boldsymbol{a}$ と定義する．この処理は，$\boldsymbol{u}^{(n)}$ に与える時間遅れを $d^{(n)}$ とおくと，次式で表せる．

$$a(l) = \sum_{n=1}^{N} u^{(n)}(l - d^{(n)}) \quad \forall l \tag{3.1}$$

式 (3.1) により，ある話者の観測音声信号 $\boldsymbol{a}$ が作成される．同様の処理を別の話者についても行い，観測音声信号 $\boldsymbol{b} \in \mathbb{R}^L$ を作成する．最後に，作成した観測音声信号 $\boldsymbol{a}$ 及び $\boldsymbol{b}$ の要素和を取ることで混合音声信号 $\boldsymbol{y} = [y(1), y(2), \cdots, y(L)]^{\mathrm{T}} \in \mathbb{R}^L$ が $\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{b}$ として作成できる．

次に，混合音声信号 $\boldsymbol{y}$ のどの時間区間が単一話者発話区間かを表す正解ラベルの作成方法について説明する．Fig. 3.3 にラベルの作成方法の概要を示す．まず，Fig. 3.3 の step 1 に示すように，観測信号 $\boldsymbol{a}$ に対して振幅方向の閾値処理を行う．具体的には，閾値以上の信号値を持つ離散時間には 1，閾値未満の離散時間には 0 付与したバイナリラベルベクトル $\boldsymbol{\alpha} = [\alpha(1), \alpha(2), \cdots, \alpha(l), \cdots, \alpha(L)]^{\mathrm{T}} \in \{0, 1\}^L$ を次式のように定義する．

$$\alpha(l) = \begin{cases} 1 & (\text{if } a(l) \geq \zeta_{|\boldsymbol{a}|}) \\ 0 & (\text{otherwise}) \end{cases} \qquad \forall l \tag{3.2}$$

ここで，閾値 $\zeta_{|\boldsymbol{a}|}$ は次式で定義する．

$$\zeta_{|\boldsymbol{a}|} = \max(|\boldsymbol{a}|) \times \delta \tag{3.3}$$

ここで，$\max(\cdot)$ は入力ベクトルの最大値を返す関数であり，δ は最大絶対振幅値に対する閾値の割合である．

次に，Fig. 3.3 の step 2 のように，時間方向の閾値処理を行う．この処理の理由は，零交差している離散時間点付近や日本語の促音等のごくわずかな無音声区間を，誤って非発話区間とラベル付けしてしまうことを回避するためである．

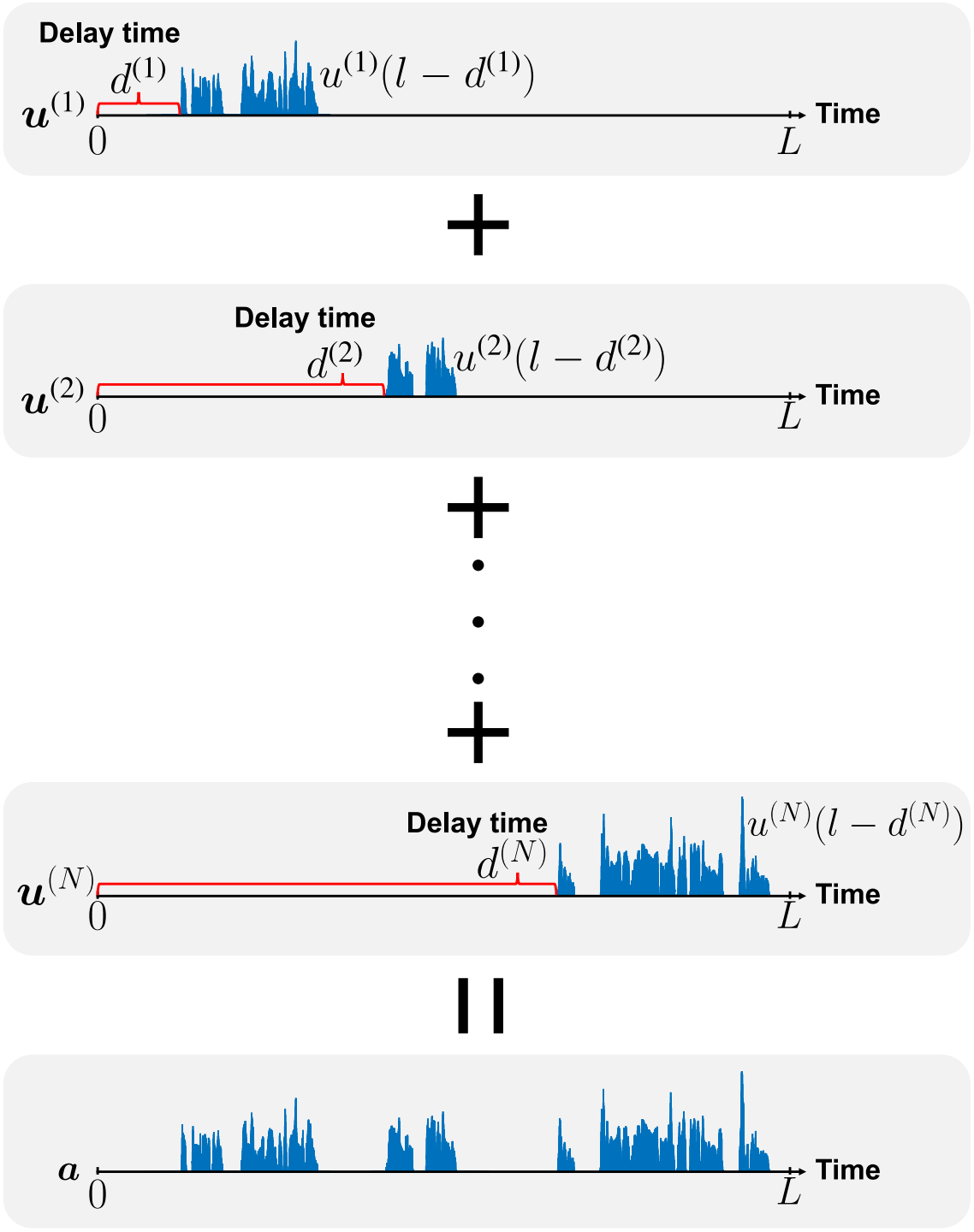


Fig. 3.2. Procedure for storing audio signals.

この処理は，Fig. 3.3 の step 2 のように，step 1 で求めたラベルベクトル $\boldsymbol{\alpha}$ の中で $\alpha(l)$ の値が連続して 0 になっている全ての時間区間に着目し，連続している 0 の個数が閾値 $\xi_{\boldsymbol{\alpha}}$ 未満であればその時間区間のラベル値を全て 1 に上書きし，閾値 $\xi_{\boldsymbol{\alpha}}$ 以上であれば変更しな

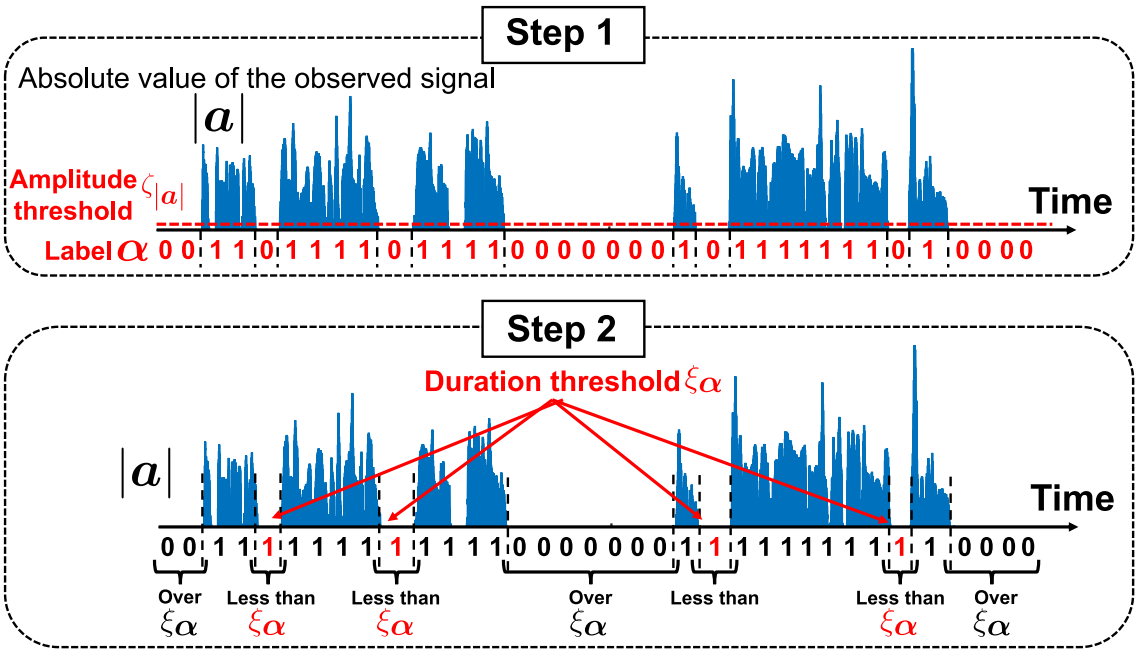


Fig. 3.3. Amplitude and time thresholding.

い，という操作となる．すなわち，$\boldsymbol{\alpha}$ の中で $\alpha(l)$ の値が連続して 0 の時間区間のラベル値を $(\alpha(l'), \alpha(l'+1), \cdots, \alpha(l'+l''-1))$ と定義（l'' は連続する 0 の個数であり，時間区間に依存して変化する定数）するとき，前述の処理は次式となる．

$$(\alpha(l'), \alpha(l'+1), \cdots, \alpha(l'+l''-1)) = \begin{cases} (1, 1, \cdots, 1) & (\text{if } l'' < \xi_{\boldsymbol{\alpha}}) \\ (\alpha(l'), \alpha(l'+1), \cdots, \alpha(l'+l''-1)) & (\text{otherwise}) \end{cases} \tag{3.4}$$

同様の方法で，別の話者のラベル $\boldsymbol{\beta}$ についても時間方向の閾値処理を施す．

最後に，二人の話者のラベル $\boldsymbol{\alpha}$ 及び $\boldsymbol{\beta}$ を用いて混合音声信号 $\boldsymbol{y}$ の正解ラベルを求める．その処理を Fig. 3.4 に示す．混合音声信号 $\boldsymbol{y}$ のラベルを $\boldsymbol{\gamma} = [\gamma(1), \gamma(2), \cdots, \gamma(l), \cdots, \gamma(L)]^{\mathrm{T}} \in \{0, 1\}^L$ と定義すると，これは単一話者発話区間に対応する離散時間インデクスでのみ 1 であり，それ以外は 0 となればよい．従って，$\boldsymbol{\alpha}$ 及び $\boldsymbol{\beta}$ を用いて次式のように求められる．

$$\gamma(l) = \begin{cases} 1 & (\text{if } \alpha(l) + \beta(l) = 1) \\ 0 & (\text{otherwise}) \end{cases} \qquad \forall l \tag{3.5}$$

以上の手続きで，混合音声信号 $\boldsymbol{y}$ の単一話者発話区間を表す正解ラベル $\boldsymbol{\gamma}$ が得られる．

提案手法では，Fig. 3.1 に示すように，混合音声信号 $\boldsymbol{y}$ を STFT して得られる振幅スペクトログラム $|\boldsymbol{Y}|$ を入力データとし，これに対応する時間フレーム毎の予測値を出力する．前述の方法で作成されたラベル $\boldsymbol{\gamma}$ は，離散時間インデクス l 毎に 0 又は 1 のバイナリ値が付されたラベルとなっている．従って，ラベル $\boldsymbol{\gamma}$ の時間解像度を落とし，STFT の時間フレーム毎の（振幅スペクトログラム $|\boldsymbol{Y}|$ の各列に対応する）バイナリ値に変換する処理を最後に施す．

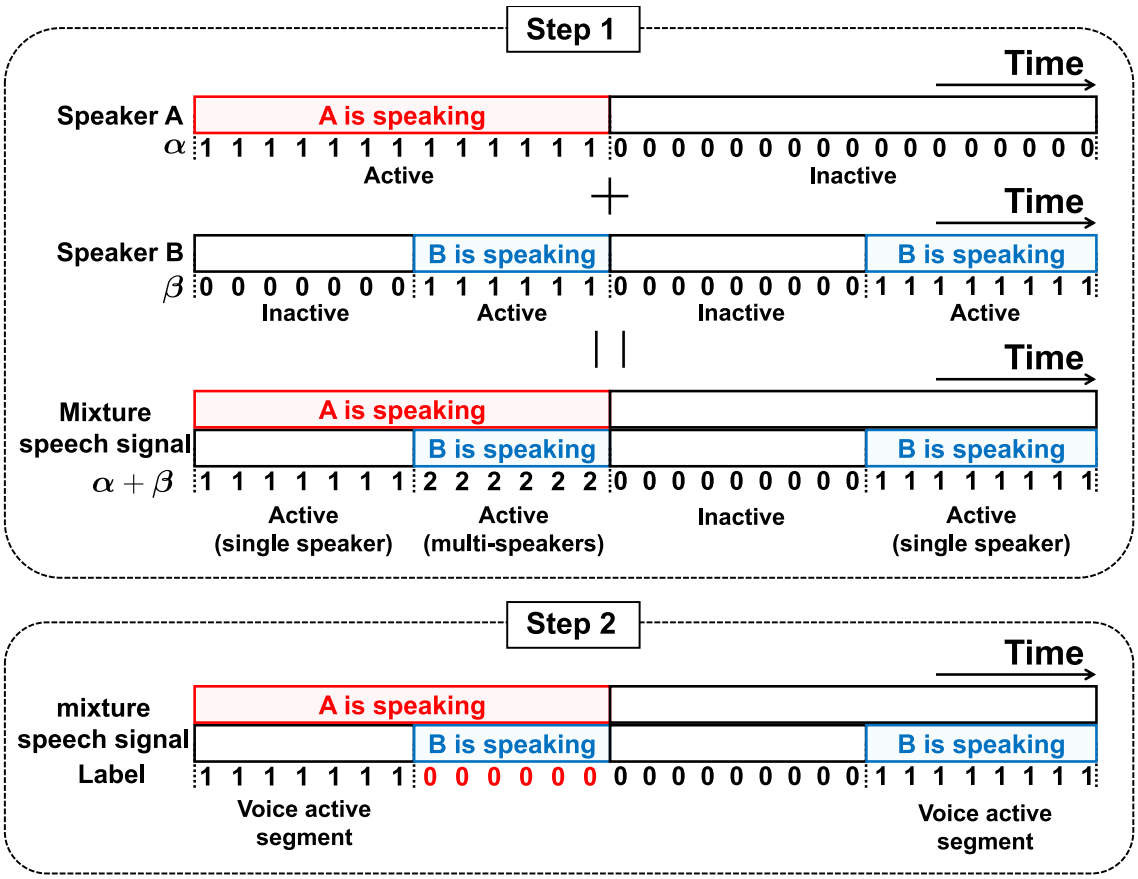


Fig. 3.4. Producing labels of single-voice active segment for mixture speech signal.

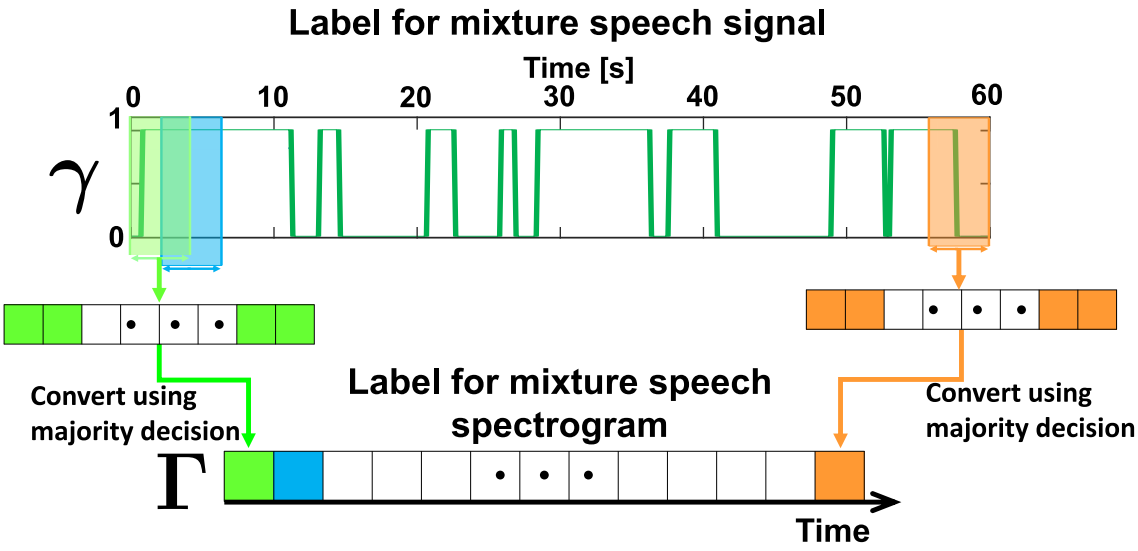


Fig. 3.5. Converting label to frame-level values.

この処理について，Fig. 3.5 に示す．具体的には，まず離散時間毎のバイナリ値を持つラベル

$\boldsymbol{\gamma}$ を，式 (2.2) と同じように窓長 Q 及びシフト長 τ を用いて J 個の短時間区間に分割する．

$$\tilde{\boldsymbol{\gamma}}^{(j)} = [\gamma((j-1)\tau+1), \gamma((j-1)\tau+2), \cdots, \gamma((j-1)\tau+Q)]^{\mathrm{T}} \tag{3.6}$$
$$= [\tilde{\gamma}^{(j)}(1), \tilde{\gamma}^{(j)}(2), \cdots, \tilde{\gamma}^{(j)}(Q)]^{\mathrm{T}} \in \mathbb{R}^{Q} \tag{3.7}$$

この短時間区間内のラベルに関して，バイナリ値 0 及び 1 の多数決処理を行い，j 番目の時間フレームの最終的なラベル $\boldsymbol{\Gamma} = [\Gamma(1), \Gamma(2), \cdots, \Gamma(j), \cdots, \Gamma(J)]^{\mathrm{T}} \in \{0,1\}^{J}$ を決定する．多数決処理をラベル閾値 ϑ と定義したとき，この処理は次式となる．

$$\Gamma(j) = \begin{cases} 1 & \left(\text{if } \frac{\sum_q \tilde{\gamma}(q)}{Q} \geq \vartheta\right) \\ 0 & (\text{otherwise}) \end{cases} \tag{3.8}$$

このようにして得られるラベル $\boldsymbol{\Gamma}$ の時間インデクス j は，振幅スペクトログラム $|\boldsymbol{Y}|$ の時間フレームと同期しているため，入力データ $|\boldsymbol{Y}|$ のラベルに用いることができる．

3.4 ネットワーク構造

本節では，提案手法の BiLSTM 及び全結合層のネットワーク構造について説明する．Fig. 3.6 に提案手法のネットワーク構造の全容を示す．入力データである振幅スペクトログラムの各時間フレームを時系列データとし，BiLSTM の順方向及び逆方向に入力する．このとき，順方向と逆方向の LSTM をそれぞれ 3 層用意する．1 層目及び 2 層目の BiLSTM の出力は，順方向と逆方向の 2 つの出力ベクトルを結合している．また，3 層目の BiLSTM の出力は順方向と逆方向の 2 つの出力ベクトルを要素毎に乗算している．1 層目の BiLSTM における各時間フレームの入力次元数は周波数ビン数に一致するため I である．その後，1 層目の BiLSTM の順方向及び逆方向処理を通した後の出力次元数は $I/4$ となるように設定している．1 層目の BiLSTM の 2 つの出力ベクトルは結合されるので，2 層目の BiLSTM の入力次元数は $I/2$ となる．2 層目の BiLSTM では，1 層目の各時間フレームの出力が Fig. 3.6 のように処理されて入力される．2 層目の BiLSTM の出力次元ベクトルの次元数は各方向で $I/16$ となり，これが再び結合されて $I/8$ となる．3 層目の BiLSTM も同様に処理され，最後に各方向の $I/64$ 次元の出力ベクトルを（結合ではなく）要素毎に乗算し，その結果を全結合層へと渡してる．全結合層は入力層と出力層の 2 層で構成され，入力層の次元数は $I/64$（BiLSTM の出力と一致），出力層の次元数は 2 と設定している．出力層の活性化関数には softmax 関数を用いているため，提案手法のネットワーク全体の最終出力は時間フレーム毎のバイナリクラスの確率値となる．これらの確率値が「単一話者発話区間である確率」及び「単一話者発話区間ではない確率」を表すようにネットワーク全体を学習する．なお，最終出力は 0 または 1 のバイナリ値ではなく確率値であるため，予測結果を得るために 2 値化を施す．具体的には，予測結果のうち「単一話者発話区間である確率」のみを集めたベクトルを $\hat{\boldsymbol{\Gamma}} = [\hat{\Gamma}(1), \hat{\Gamma}(2), \cdots, \hat{\Gamma}(j), \cdots, \hat{\Gamma}(J)]^{\mathrm{T}} \in [0,1]^{J}$ と定義するとき，次式を計算することで 2 値

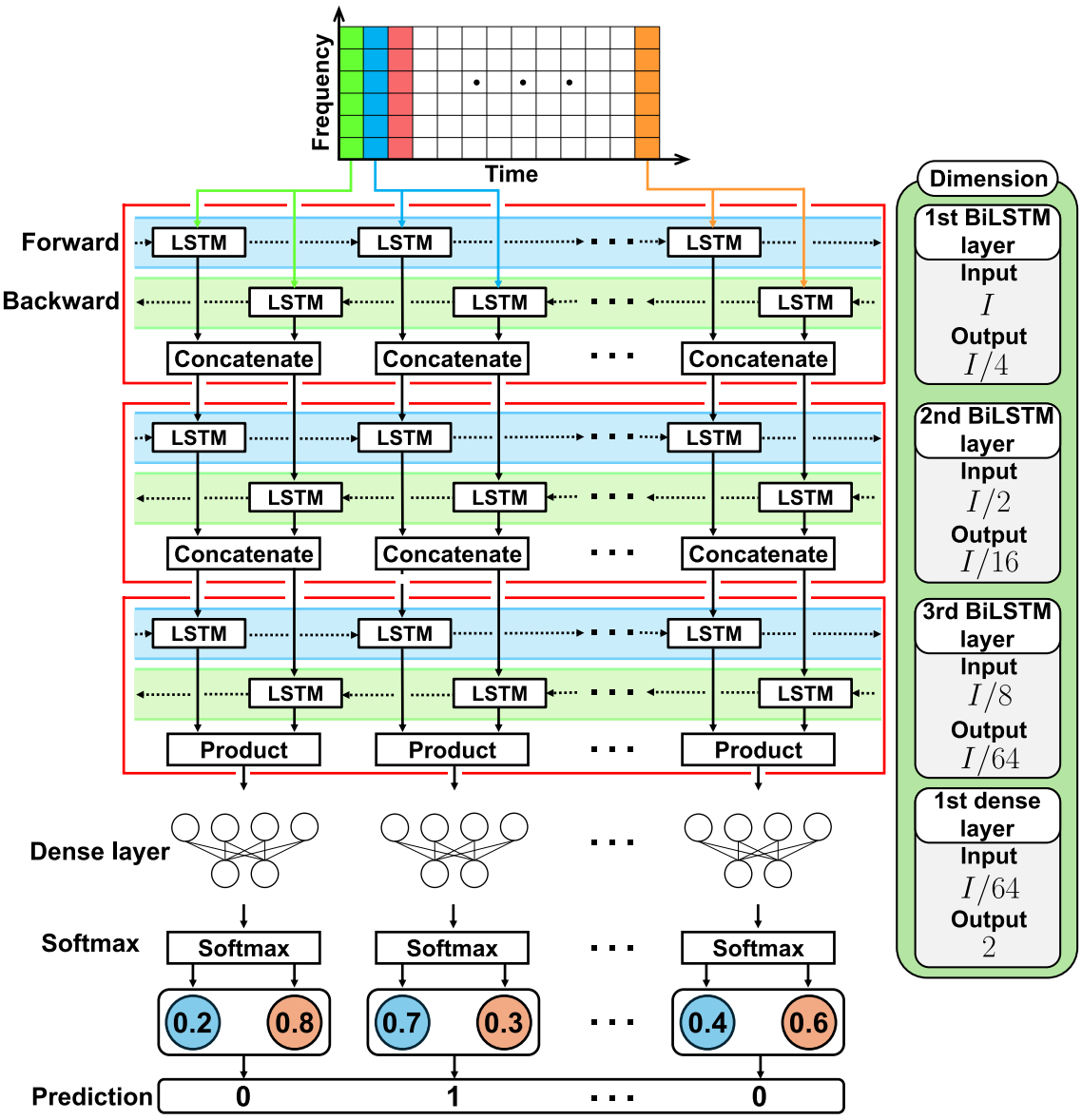


Fig. 3.6. Network architecture of proposed method.

化する．

$$\hat{\Gamma}^{(\mathrm{bin})}(j) = \begin{cases} 1 & (\text{if } \hat{\Gamma}(j) \geq 0) \\ 0 & (\text{otherwise}) \end{cases} \quad \forall j \tag{3.9}$$

3.5　ネットワークの学習

本節では提案手法のネットワークの学習に関する説明を行う．まず，本実験の深層学習では学習データ，検証データ，及びテストデータの 3 種類を準備する．このとき，混合音声信号内で発話している話者は学習データ，検証データ，及びテストデータで異なる話者となるように

混合音声信号を作成する．

次にネットワークのパラメータの学習（更新）に必要な損失の計算方法について説明する．損失の計算は，2 値化する前の予測結果ベクトル $\hat{\mathbf{\Gamma}}$ とラベルベクトル $\mathbf{\Gamma}$ 間のバイナリクロスエントロピー（binary cross entropy: BCE）損失を計算する．BCE 損失は次式で表される．

$$\mathcal{L}_{\mathrm{BCE}}(\mathbf{\Gamma}, \hat{\mathbf{\Gamma}}) = -\frac{1}{J}\sum_{j=1}^{J}\left[\Gamma(j)\log\hat{\Gamma}(j) + (1-\Gamma(j))\log(1-\hat{\Gamma}(j))\right] \tag{3.10}$$

学習ステージでは，学習データを用いてこの BCE 損失が小さくなるように誤差逆伝播を用いてネットワーク全体のパラメータを更新していく．過学習を防ぐために，検証データに対する予測性能をモニタリングし，早期終了を適用する．最終的に学習済みのネットワークを用いてテストデータに対する単一話者発話区間を予測し，提案手法の性能を評価する．

3.6 本章のまとめ

本章では，1 章で説明した提案手法について説明し，実験に用いる混合音声信号，正解ラベルの作成方法，及び提案手法のネットワーク構造について説明した．3.2 節では DNN に基づく SVAD の詳しい説明を行った．具体的な実験条件等は 4 章で説明する．3.3 節では実験に用いる混合音声信号と正解ラベルの作成手順について数式を交えて説明した．3.4 節では実験で用いる BiLSTM のネットワーク情報について説明した．3.5 節では，ネットワークの具体的な学習手順について説明した．4 章では，提案手法の実験条件と実験方法及び実験結果について説明し，得られた結果をもとに結果の解説を行う．

第 4 章

単一話者発話区間の推定実験

4.1　まえがき

本章では，3 章で述べた提案手法を用いて，単一話者発話区間を予測する実験の詳細とその結果について説明する．4.2 節で実験条件について説明する．4.3 節で提案手法の学習及び評価を行い，その性能について議論する．4.4 節で本章をまとめる．

4.2　実験条件

本実験では，3 章で述べた提案手法がどの程度の精度で単一話者発話区間を検出できるか，その精度を評価する．本節では，本実験の具体的な実験条件に付いて網羅的に説明する．

まず，本実験で使用する混合音声信号は，JVS corpus を使用して，3.3 節の方法で作成した．JVS corpus には，日本人男女 50 名ずつの計 100 名のデータが含まれており，各話者に対して 100 種類のパラレル発話データ（同じ文章を読み上げているデータ），30 種類のノンパラレル発話データ，10 種類のささやき声の発話データ，及び 10 種類の裏声の発話データが含まれている．本実験では，男女 36 名の話者からそれぞれ 30 種類のノンパラレル発話データを抽出し，2 名の話者の発話を混合することで，発話者の異なる 18 組の混合音声信号を計 1800 種類作成した．1 つの混合音声信号の信号長は 60 s とした．学習データは 1 組あたり 150 個の混合音声信号を 8 組，検証データはは 1 組あたり 60 個の混合音声信号を 6 組，及びテストデータは 1 組あたり 60 個の混合音声信号を 5 組準備した．組が異なれば話者も全く異なるため，学習データ，検証データ，及びテストデータは全て互いに同一話者を含まないような分割となっている．なお，これらの作成した混合音声信号のサンプリング周波数は全て 16 kHz に統一している．その他入力信号に関する実験条件を Table 4.1 に示す．なお，式 (3.8) における多数決処理の割合 ϑ は予測結果に強く影響することが予想されるため，Table 4.1 に示す 3 種類の値で実験しその性能の変化についても確認する．

続いて，提案手法のネットワークの学習条件を説明する．学習のエポック数は 500 回に設定した．また，過学習を防ぐため検証データの損失が過去の値より 15 回以上改善されない段階

Table 4.1. Experimental conditions for input signals

Parameter	Value
Amplitude threshold rate (δ)	0.004
Duration threshold ($\xi_{\boldsymbol{\alpha}}$ and $\xi_{\boldsymbol{\beta}}$)	8000 samples (500 ms)
Window length in STFT (Q)	4096 samples (256 ms)
Shift length in STFT (τ)	2048 samples (128 ms)
Label threshold (ϑ)	0.024%/50.000%/99.976%

で早期終了をかけた．このとき，パラメータは検証データの損失がもっとも低い時の値を保存する．さらに，最適化アルゴリズムには Adam [29] を使用し，その学習率は 0.01 に設定した．

4.3 実験結果

Figs. 4.1 (a)，(b)，及び (c) にそれぞれ，ラベル閾値を $\vartheta = 0.500$ と設定した場合の学習中の損失の値の推移，正解率の推移，及びテストデータにおける予測結果とラベルの一例を示す．但し，Fig. 4.1 (c) は合計 300 個あるテストデータからランダムに選んだ 1 つの予測結果例である．また，予測結果とラベルは本来 0 又は 1 の 2 値で表される数値であるが，Fig. 4.1 (c) では見やすくするために，予測結果とラベルの値をそれぞれ 0.4 倍及び 0.42 倍して表示している（すなわち，1 の値がそれぞれ 0.4 及び 0.42 に対応している）．

まず，Fig. 4.1 (a) の損失のグラフを見ると，エポック数が 33 の時点で学習が早期終了していることが分かる．学習データの損失は早期終了するまで下がり続けており，早期終了時の損失値は 0.0169 であった．一方，検証データの損失は最も低い時で 0.282 となった．次に，Fig. 4.1 (b) の正解率のグラフを見ると，学習データの正解率は早期終了時にほぼ 100% となっていることが分かる．検証データの正解率は早期終了時に 88.526% となり，1.2 節で述べた BSS への応用に提案手法を活用することを考えると，十分な精度での予測ができていると考えられる．

この学習結果の中で，検証データの損失が最も低かったエポック数 18 の時点でのネットワークを保存し，テストデータにに適用した結果，300 個のテストデータ全体で 87.473% の予測精度を達成した．この結果も，我々の期待する BSS への応用には十分な予測精度と考えられる．テストデータの予測結果の一例を Fig. 4.1 (c) に示している．ここで，このデータでの予測の正解率は 90.176 % であり，殆どの時間区間で単一話者発話区間の推定が正しくできているといえる．

しかし，Fig. 4.1 (c) の 0 s から 10 s の予測とラベルを比較すると，1〜2 秒の短い非単一話者発話区間の推定が正確にできていないことが分かる．この理由として，この予測を誤った時間区間は両話者が沈黙している区間であるが，その時間長が短すぎることが原因でうまく予測できなかったものと考えられる．さらに細かく見ると，予測の単一話者発話区間の開始点や終

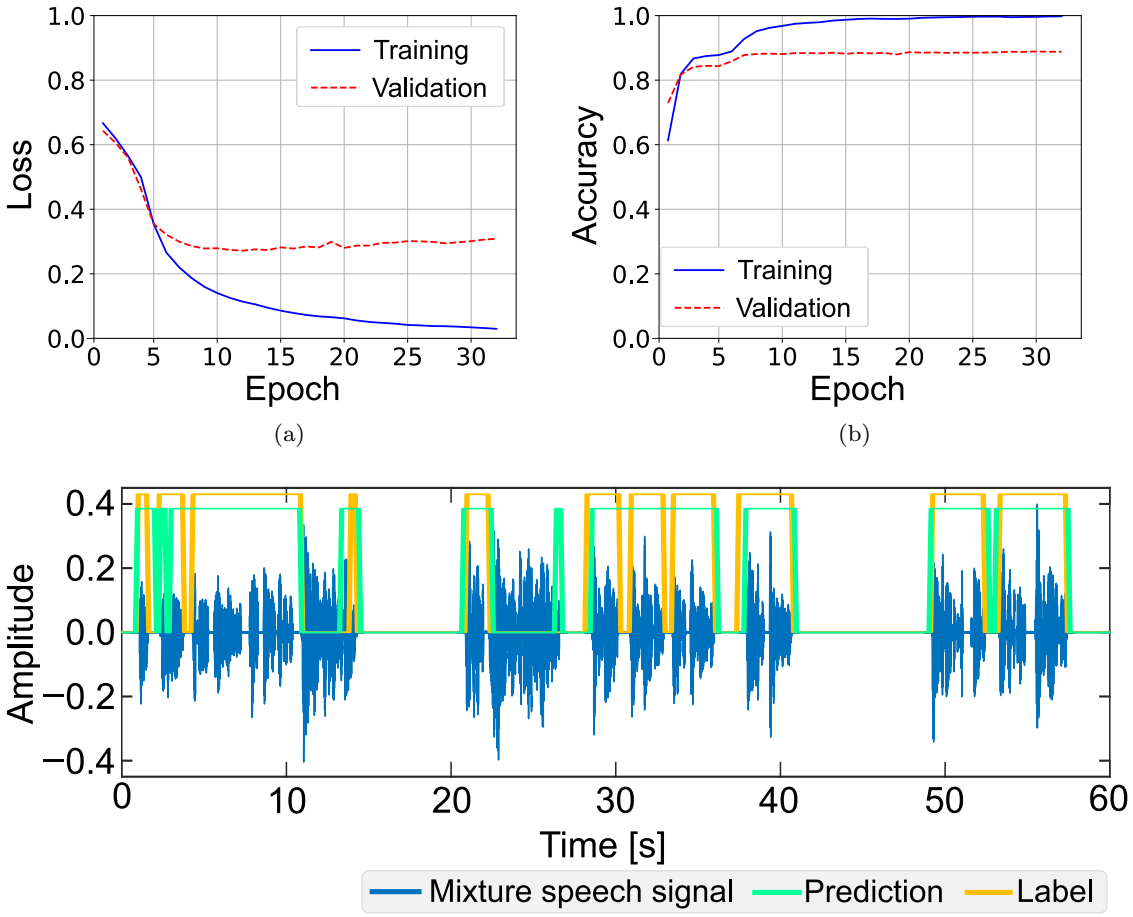


Fig. 4.1. Experimental result when $\vartheta = 50.000\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.

了点は，ラベルと比較して数百 ms 程度のずれが生じていることが，10 s から 20 s の予測とラベルを見るとわかる．このずれの考えられる原因として，発話区間の時間長が短すぎるため予測がうまくいかず，予測の単一話者発話区間の開始点及び終了点にずれが生じることがあげられる．発話区間及び沈黙区間の時間長が短い区間の予測があまりうまくいかないことが Figs. 4.1 (a)，(b)，及び (c) の結果よりわかった．

ここでは $\vartheta = 0.00024$ の時の説明を行う．Figs. 4.2 (a)，(b)，及び (c) にそれぞれ，ラベル閾値を $\vartheta = 0.00024$ と設定した場合の学習中の損失の値の推移，正解率の推移，及びテストデータにおける予測結果とラベルの一例を示す．但し，Fig. 4.2 (c) は合計 300 個あるテストデータからランダムに選んだ 1 つの予測結果例である．また，予測結果とラベルは本来 0 又は 1 の 2 値で表される数値であるが，Fig. 4.2 (c) では見やすくするために，予測結果とラベルの値をそれぞれ 0.4 倍及び 0.42 倍して表示している（すなわち，1 の値がそれぞれ 0.4 及び 0.42 に対応している）．

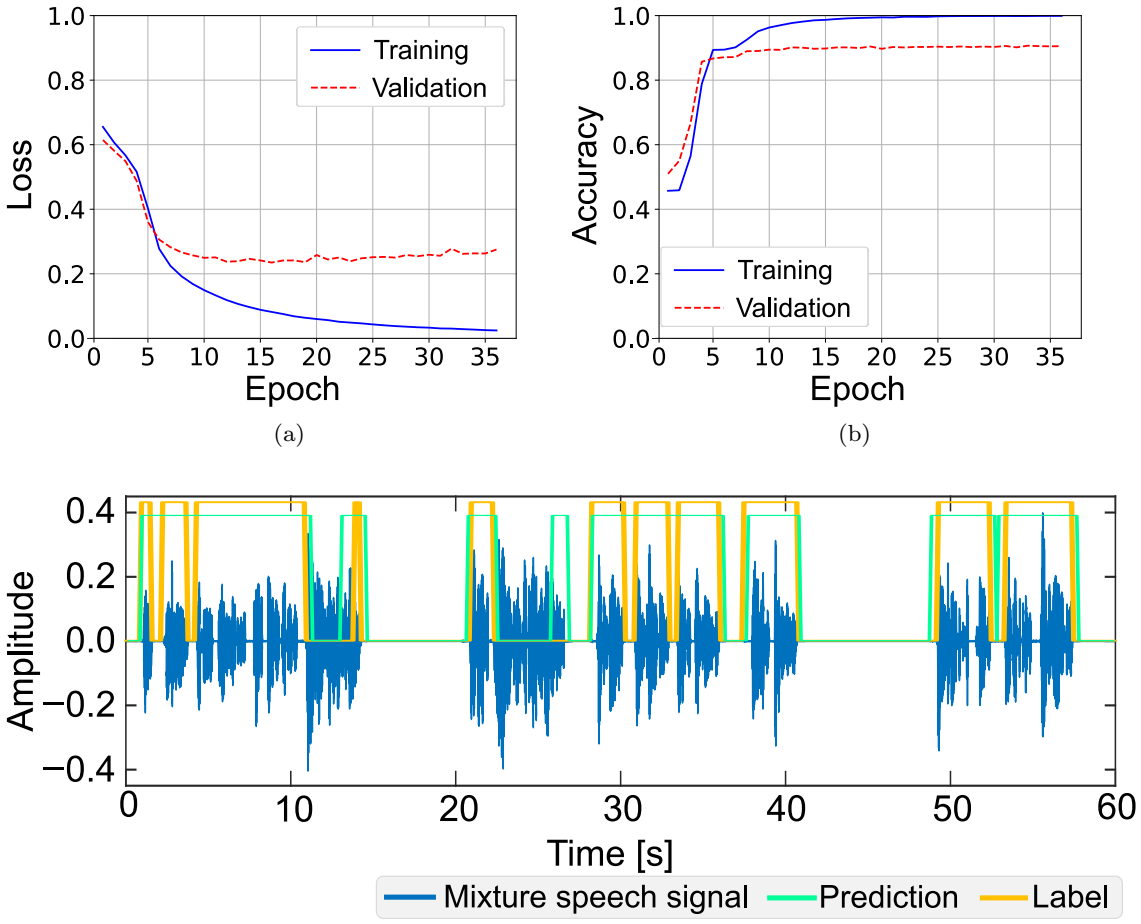
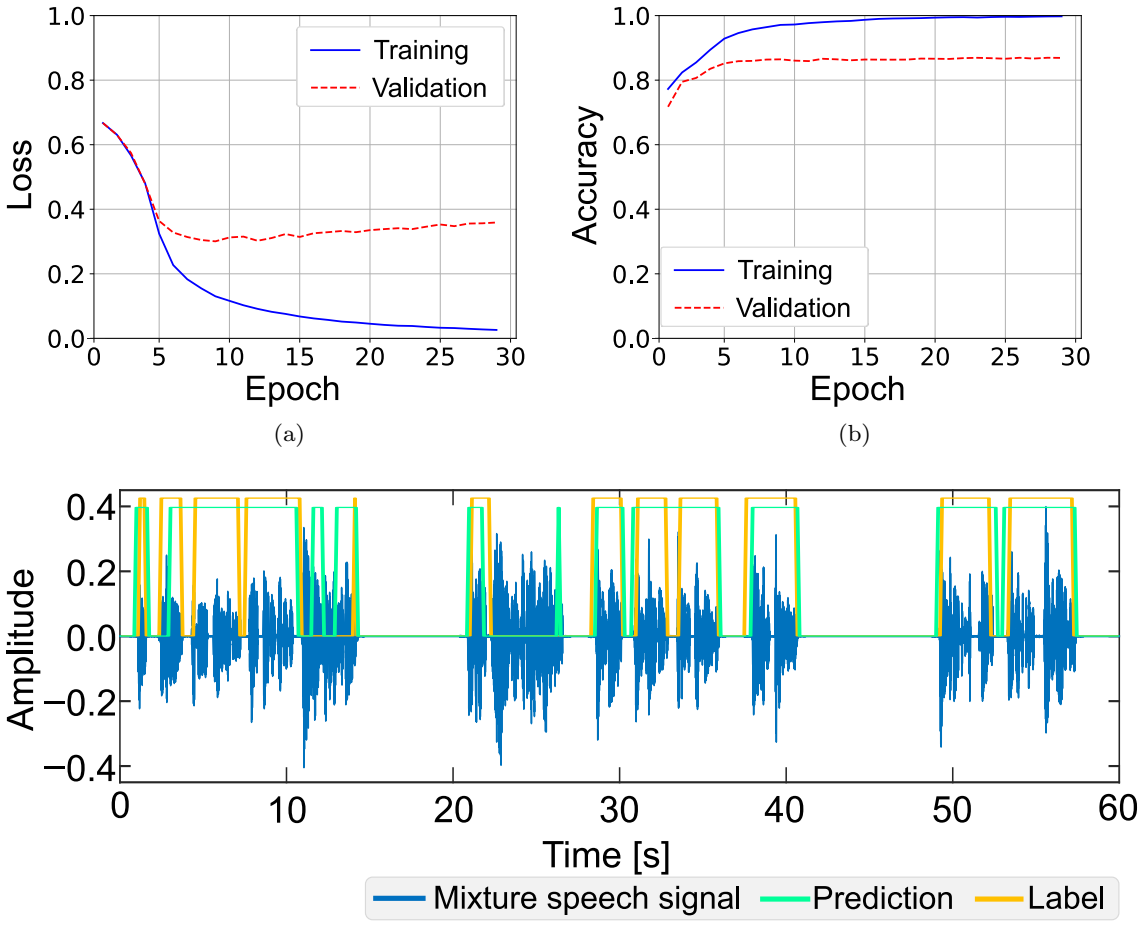


Fig. 4.2. Experimental result when $\vartheta = 0.024\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.

まず，Fig. 4.2 (a) の損失のグラフを見ると，エポック数が 36 の時点で学習が早期終了していることが分かる．学習データの損失は早期終了するまで下がり続けており，早期終了時の損失値は 0.0213 であった．一方，検証データの損失は最も低い時で 0.247 となった．次に，Fig. 4.2 (b) の正解率のグラフを見ると，学習データの正解率は早期終了時にほぼ 100% となっていることが分かる．検証データの正解率は早期終了時に 91.114% となり，1.2 節で述べた BSS への応用に提案手法を活用することを考えると，十分な精度での予測ができていると考えられる．

この学習結果の中で，検証データの損失が最も低かったエポック数 21 の時点でのネットワークを保存し，テストデータにに適用した結果，300 個のテストデータ全体で 90.593% の予測精度を達成した．この結果も，我々の期待する BSS への応用には十分な予測精度と考えられる．テストデータの予測結果の一例を Fig. 4.2 (c) に示している．ここで，このデータでの予測の正解率は 92.186 % であり，殆どの時間区間で単一話者発話区間の推定が正しくできて



(a)　(b)

(c)

Fig. 4.3. Experimental result when $\vartheta = 99.976\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.

いるといえる．

しかし，Fig. 4.2 (c) の 0 s から 10 s の予測とラベルを比較すると，1〜2 秒の短い非単一話者発話区間の推定が正確にできていないことが分かる．この理由として，Fig. 4.1 (c) と同様に沈黙区間の時間長が短すぎることが原因でうまく予測できなかったものと考えられる．さらに細かく見ると，予測の単一話者発話区間の開始点や終了点は，ラベルと比較して数百 ms 程度のずれが生じていることが，10 s から 20 s の予測とラベルを見るとわかる．Fig. 4.1 (c) の正解率を比較すると，発話区間及び沈黙区間の時間長が短い区間の予測があまりうまくいっていないが，Fig. 4.2 (c) の正解率のほうが高いことがわかる．この理由としてラベル閾値を ϑ の値を 0.500 から 0.00024 に変化させたことによりラベルの多数決で発話区間と判別されやすくなったため沈黙区間の割合が下がり結果的に正解率の増加につながった．

ここでは $\vartheta = 0.99976$ の時の説明を行う．Figs. 4.3 (a)，(b)，及び (c) にそれぞれ，ラベル閾値を $\vartheta = 0.99976$ と設定した場合の学習中の損失の値の推移，正解率の推移，及びテスト

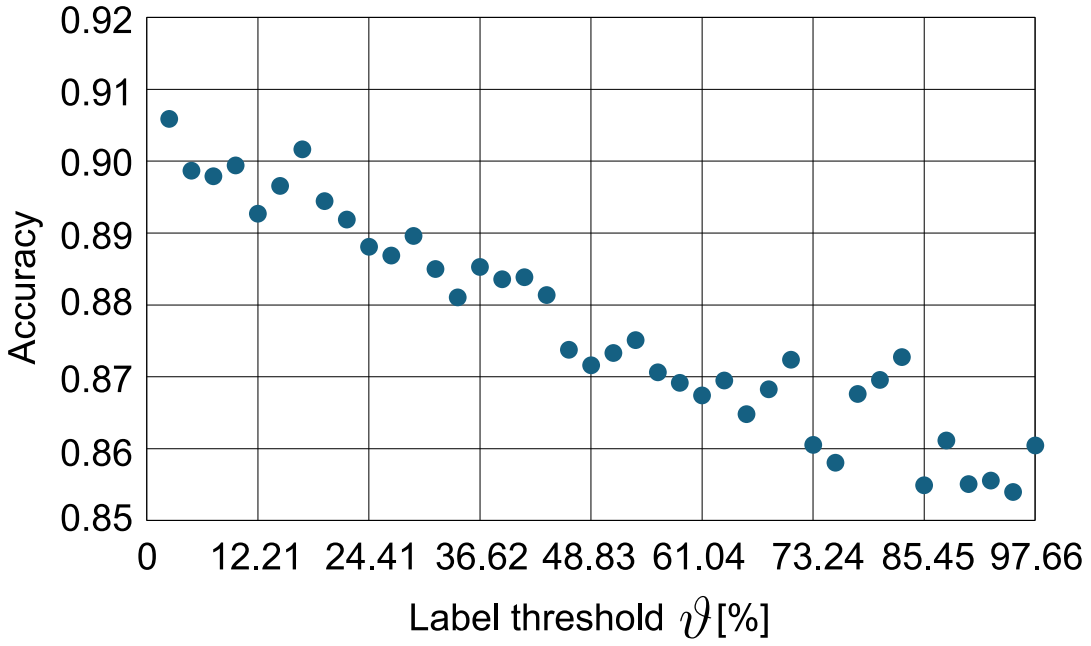


Fig. 4.4. Prediction accuracy with various label threshold values.

データにおける予測結果とラベルの一例を示す．但し，Fig. 4.3 (c) は合計 300 個あるテストデータからランダムに選んだ 1 つの予測結果例である．また，予測結果とラベルは本来 0 又は 1 の 2 値で表される数値であるが，Fig. 4.3 (c) では見やすくするために，予測結果とラベルの値をそれぞれ 0.4 倍及び 0.42 倍して表示している（すなわち，1 の値がそれぞれ 0.4 及び 0.42 に対応している）．

まず，Fig. 4.3 (a) の損失のグラフを見ると，エポック数が 29 の時点で学習が早期終了していることが分かる．学習データの損失は早期終了するまで下がり続けており，早期終了時の損失値は 0.0234 であった．一方，検証データの損失は最も低い時で 0.316 となった．次に，Fig. 4.3 (b) の正解率のグラフを見ると，学習データの正解率は早期終了時にほぼ 100% となっていることが分かる．検証データの正解率は早期終了時に 86.574% となり，1.2 節で述べた BSS への応用に提案手法を活用することを考えると，十分な精度での予測ができていると考えられる．

この学習結果の中で，検証データの損失が最も低かったエポック数 21 の時点でのネットワークを保存し，テストデータにに適用した結果，300 個のテストデータ全体で 86.131% の予測精度を達成した．この結果は，我々の期待する BSS への応用に十分な予測精度と考えられる．テストデータの予測結果の一例を Fig. 4.3 (c) に示している．ここで，このデータでの予測の正解率は 86.759 % であり，殆どの時間区間で単一話者発話区間の推定が正しくできているといえる．Fig. 4.2 (c) の 0 s から 10 s の予測とラベルを比較すると，1〜2 秒の短い非単一話者発話区間の推定ができているが，開始点及び終了点のずれが大きく，正確ではないことが分かる．ただし，30 s から 40 s の予測とラベルを比較すると，一部時間区間の 1〜2 秒の短い非単一話者発話区間の推定が正確にできていることが分かる．この理由として，Fig. 4.3

(c) ではラベル閾値を $\vartheta = 0.99976$ にしたことにより沈黙区間の時間長が広がりうまく予測できたと考えられる．

Fig. 4.4 にラベル閾値 ϑ を 100 から 4000 に変化させたときのテストデータの正解率を示す．図中の各プロットの正解率は全てのテストデータの正解率の平均値を示す．正解率の推移をみると，ϑ の値を増加させるにつれて正解率は減少していることがわかる．その理由としてラベル閾値 ϑ の値が大きくなるほど沈黙の短い区間の推定がうまくできるが，発話区間の開始及び終了地点のずれが大きくなるからと考えられる．

4.4　本章のまとめ

本章では，3 章で説明した提案手法について推定実験を行い，テストデータに対する推定精度の評価を行った．4.2 節では実験に用いる混合音声信号の詳細，実験条件を示した．4.3 節では提案手法がどの程度の精度で単一話者発話区間を検出できるか，その精度を評価した．結果として 90% を超える正答率になることを示した．5 章では，本論文における結論を述べる．

第 5 章

結言

本論文では，BSS の分離精度向上を目的とした，単一話者発話区間検出器である SVAD を新たに提案した．1 章では，BSS に関する研究を紹介し，本論文の目的について述べた．2 章では，実験に用いる要素技術について説明した．提案手法の類似研究である VAD 及び話者ダイアライゼーションの紹介を行った．3 章で，提案手法について詳しく説明した．SVAD を実現する手法として BiLSTM を用いた手法を提案し，具体的な単一話者発話区間の検出方法について説明した．提案手法を学習するための入力信号（混合音声信号）とそのラベル（単一話者発話区間情報）の作成方法について述べた．4 章で，学習に用いるデータセットを準備し，実際に単一話者発話区間の推定を行った．学習データ，及び検証データでは損失関数の値をもとにパラメータである重みの設定を行い，テストデータで学習済みモデルの汎化性を評価した．更に，予測精度に影響すると考えられるラベル閾値 ϑ の割合を変更し，精度の比較を行った．推定実験の結果としては，ラベルに対する予測の精度は BSS への応用に提案手法を活用することを考えると，十分な精度での予測ができていることが実験結果から分かった．

最後に今後の課題を述べる．検証データにおける損失が 3 割程度しか下がらず，正解率も 90% 程度であったため，データセットの改良，ネットワーク構造の改善が必要である．更に本実験で使用した混合音声信号は 2 人の話者であったが，3 人以上の音声が含まれる混合音声信号に対しても実験を行う必要がある．

謝辞

本論文は，香川高等専門学校電気情報工学科北村研究室にて行われた研究に基づくものです．論文を完成させるにあたり，多くの方々に支えられ，助言を頂きました．その中で特に以下の方々に深く感謝の意を表します．

まず，本研究を進めるにあたり，指導教員である北村大地講師には，ご多忙のところ専念したご指導と熱心なアドバイスに心から感謝申し上げます．北村大地講師の専門的な知識と卓越した指導のおかげで，私は新しい視点から研究課題に取り組むことができ，その結果，深い理解と充実感を得ることができました．北村大地講師が提供してくださったリソースやサポートは，私の研究をより効果的かつ意義深いものに仕上げる一助となりました．心よりの感謝を申し上げます．

本論の副査である雛元洋一助教には，論文の構成や記述に関して大変有益な助言を頂き，大変お世話になりました．ここに厚く御礼申し上げます．

次に，北村研究室の先輩である専攻科 2 年の川口翔也氏，溝渕悠朔氏，村田佳斗氏，専攻科 1 年の綾野翔馬氏には，研究を進めるにつき必要な基礎技術のご説明をはじめ，プログラミング等知識，研究の進め方など様々なご支援をいただきました．まず，先輩の熱心な助言と経験豊富な知識に感謝いたします．また，専攻科 2 年の川口翔也氏には，DNN に関するアドバイスやプログラミングに関する経験豊富な知識をはじめ，様々なご支援，ご助言をいただき，それが私の研究にポジティブな影響をもたらしました．心より感謝申し上げます．

また，北村研究室同期の鈴木慶氏，松本愛花氏，和気佑弥氏には，日頃のディスカッションのほか，多くの支えと励ましをいただきました．まず，同期の皆様には研究室内外での協力と助け合いに感謝いたします．共に研究室で過ごした日々は，私の成長と学びに満ちたものとなりました．1 年に亘る研究室生活を様々な面で支えていただきました．心からの感謝を込めて，ありがとうございました．

最後になりますが，私の人生において最も大きな支えとなってくれた両親に不覚感謝の意を表します．両親の理解がなければ，私が研究の道へ進むこともなかったでしょう．まず初めに，現在に至るまで私の学生生活を金銭的に支え，絶え間ない励ましとサポートをくれた両親に心から感謝いたします．両親への感謝の気持ちは言葉では言い表せないほどですが，これからも感謝の気持ちを胸に，両親に恩返しできるよう一生懸命努力し続けます．

最後に，これらの方々からのご支援があって初めてこの論文を仕上げることができました．心より感謝を申し上げます．

参考文献

[1] S. Makino, S. Araki, R. Mukai, and H. Sawada,"Audio source separation based on independent component analysis," *Proc. IEEE International Symposium on Circuits and Systems*, vol. 5, 2004.

[2] H. Sawada, N. Ono, H. Kameoka, D. Kitamura, and H. Saruwatari, "A review of blind source separation methods: Two converging routes to ILRMA originating from ICA and NMF," *APSIPA Trans. Signal and Information Processing*, vol. 8, no. e12, pp. 1–14, 2019.

[3] V. Zarzoso, R. Phlypo, and P. Comon, "A contrast for independent component analysis with priors on the source kurtosis signs," *IEEE Signal Processing Letters*, vol. 15, pp. 501–504, 2008.

[4] T. Kim, H. T. Attias, S.-Y. Lee, and T.-W. Lee, "Blind source separation exploiting higher-order frequency dependencies," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 70–79, 2007.

[5] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1626–1641, 2016.

[6] S. Mogami, N. Takamune, D. Kitamura, H. Saruwatari, Y. Takahashi, K. Kondo and N. Ono, "Independent low-rank matrix analysis based on time-variant sub-Gaussian source model for determined blind source separation," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 28, pp. 503–518, 2020.

[7] C. J. Hendriksz, P. Harmatz, M. Beck, S. Jones, T. Wood, R. Lachman, C.G. Gravance, T. Orii, and S. Tomatsu, "Review of clinical presentation and diagnosis of mucopolysaccharidosis IVA," *Molecular Genetics and Metabolism*, vol. 110, pp. 54-–64, 2013.

[8] T. Kim, "Real-time independent vector analysis for convolutive blind source separation," *IEEE Trans. Circuit and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1431–1438, 2010.

[9] D. Kitamura and K. Yatabe, "Consistent independent low-rank matrix analysis for

determined blind source separation," *EURASIP J. Advances in Signal Processing*, vol. 2020, no. 46, 2020.

[10] J. Wang, S. Guan, S. Liu, and X.-L. Zhang, "Minimum-volume multichannel nonnegative matrix factorization for blind audio source separation," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 29, pp. 3089–3103, 2021.

[11] J. Gua, L. Cheng, D. Yao, J. Li, Y. Yana, "The effect of source sparsity on independent vector analysis for blind source separation," *Signal processing*, vol. 213, no. 109199, 2023.

[12] 草水 智浩, 山本 一公, 北岡 教英, 中川 聖一, "VAD が音声認識性能に与える影響," *FIT2007*（第 *6* 回情報科学技術フォーラム）, vol. 6, no. 2, pp. 269–270, 2007.

[13] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1557–1565, 2006.

[14] K. Greff, R. K. Srivastava, J. Koutn ´ık, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[15] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, 1999.

[16] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain, "Improving speaker diarization," *hal-01451540*, pp.1–5, 2004.

[17] 石塚 健太郎, 藤本 雅清, 中谷 智広, "音声区間検出技術の最近の研究動向," **日本音響学会誌**, vol. 65, no. 10, pp. 537–543, 2009.

[18] L. R. Rabiner and M. R. Sambur, "An algorithm for determining the endpoints of isolated utterances," *Bell System Technical Journal*, vol. 54, no. 2, pp. 297–315, 1975.

[19] M. Sharma, S. Joshi, T. Chatterjee, and R. Hamid, "A comprehensive empirical review of modern voice activity detection approaches for movies and TV shows," *Neurocomputing*, vol. 494, pp. 116–131, 2022.

[20] T. J. Park, N. R. Koluguri, J. Balam, and B. Ginsburg, "Multi-scale speaker diarization with dynamic scale weighting," *Proc. Interspeech*, pp. 5080–5084, 2022.

[21] M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 217–227, 2013.

[22] G. Sell, D. Snyder, A. McCree, D. Garcia–Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, and S. Watanabe, "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," *Proc. Interspeech*, vol. 2018, pp. 2808–2812, 2018.

[23] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-

to-end neural speaker diarization with selfattention," *IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 296–303, 2019.

[24] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain, "Multi-stage speaker diarization of broadcast news," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1505–1512, 2006.

[25] D. G. Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," *IEEE International Conference on Acoustic, Speech, and Signal Processing*, pp. 4930–4934, 2017.

[26] F. Jia, S. Majumdar, and B. Ginsburg, "Marblenet: Deep 1d time-channel separable convolutional neural network for voice avtivity detection," *arXiv: 2010.13886*, 2021.

[27] N. R. Koluguri, T. Park, and B. Ginsburg, "Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context," *IEEE International Conference on Acoustic, Speech, and Signal Processing*, pp. 8102–8106, 2022.

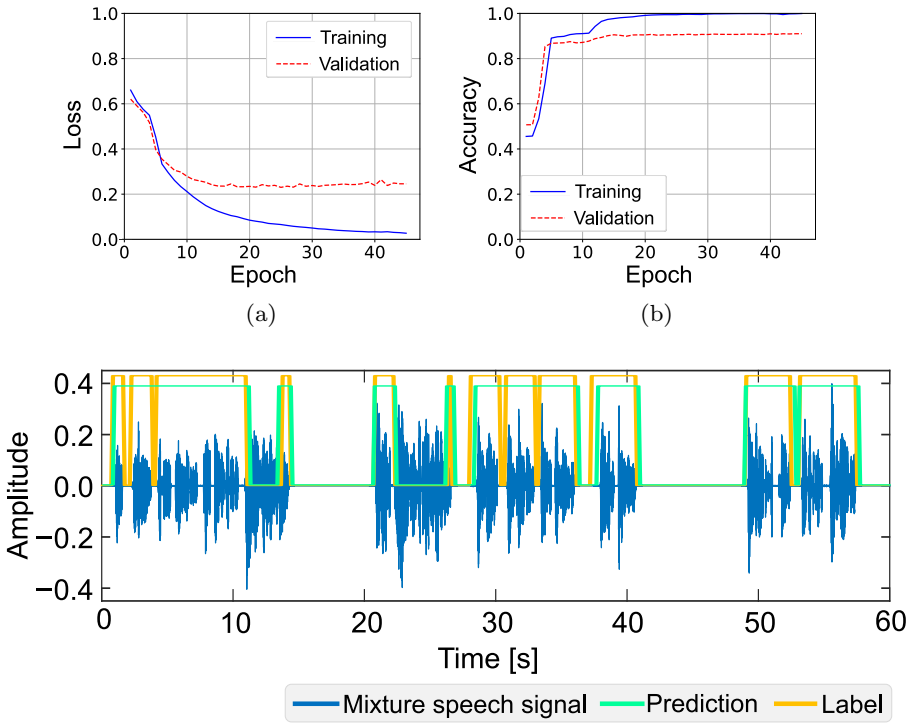
[28] S. Takamichi, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji, and H. Saruwatari, "JVS corpus: free Japanese multi-speaker voice corpus," *arXiv: 1908.06248v1*, 2019.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv: 1412.6980*, 2014.

付録 A

単一話者発話区間の推定実験

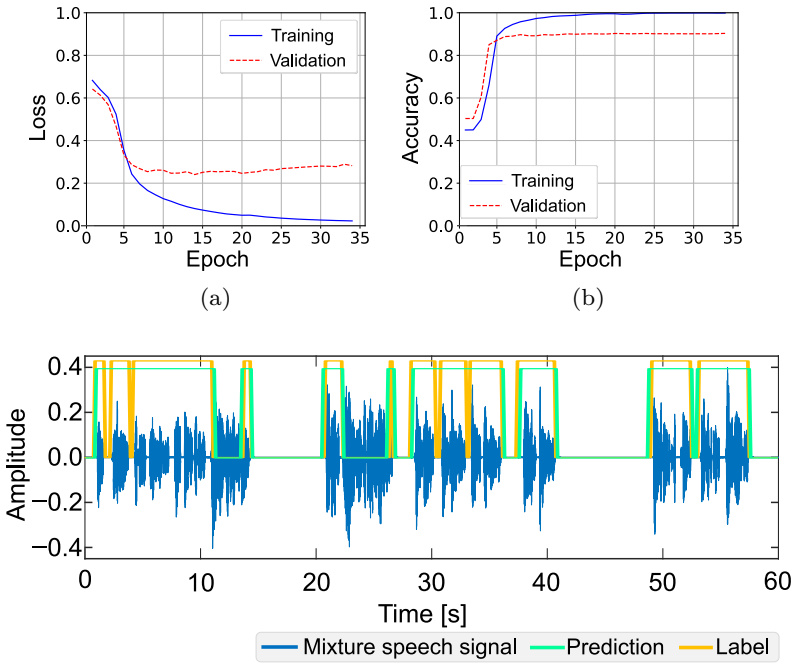
本付録では，本文で掲載しなかった残りの結果についてまとめて掲載する．Figs. A.1–A.40 にラベル閾値 ϑ の値を変化させた際の損失，正解率，及びテストデータの予測とラベルの一例をそれぞれ (a)，(b)，(c) に示す．図中の表示に関しては，Figs. 4.2–4.4 と同様である．



(a) (b)

(c)

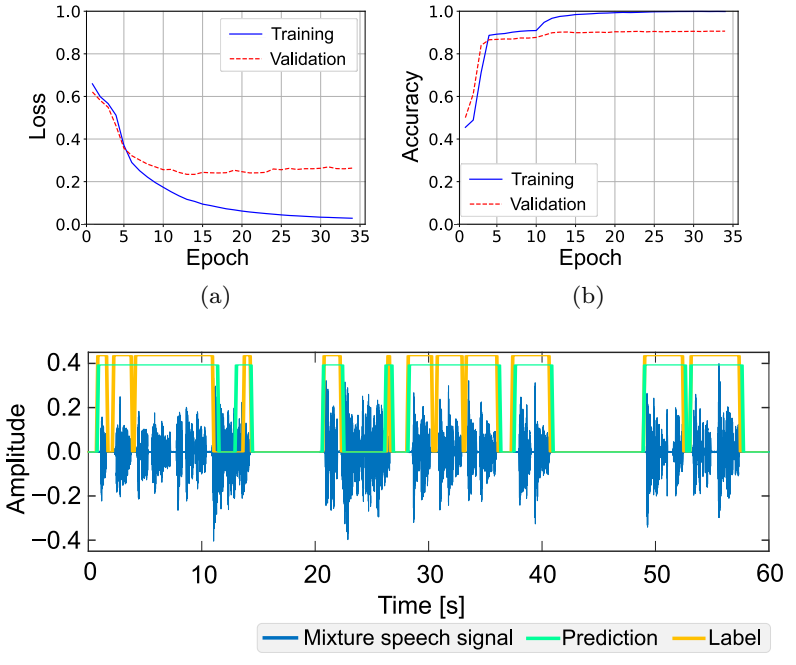
Fig. A.1. Experimental result when $\vartheta = 2.441\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

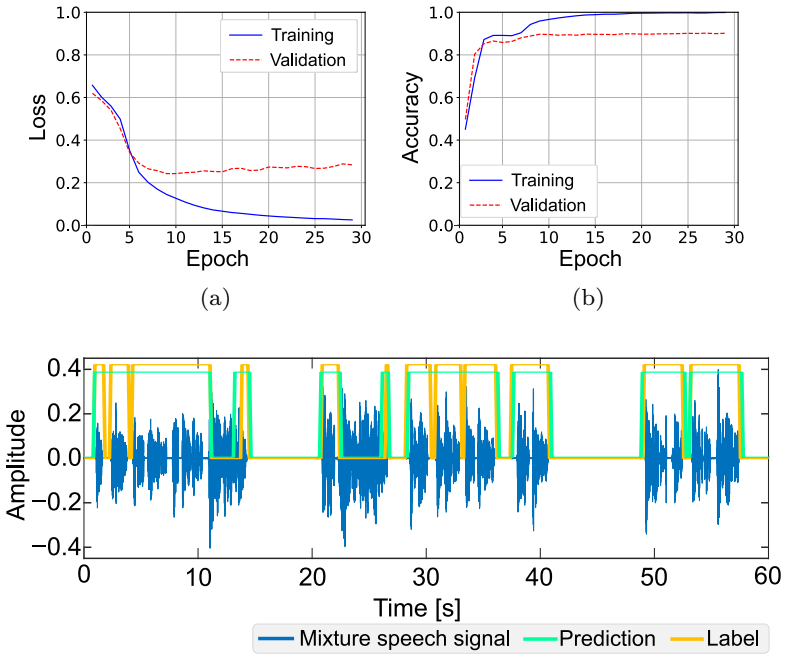
Fig. A.2. Experimental result when $\vartheta = 4.883\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

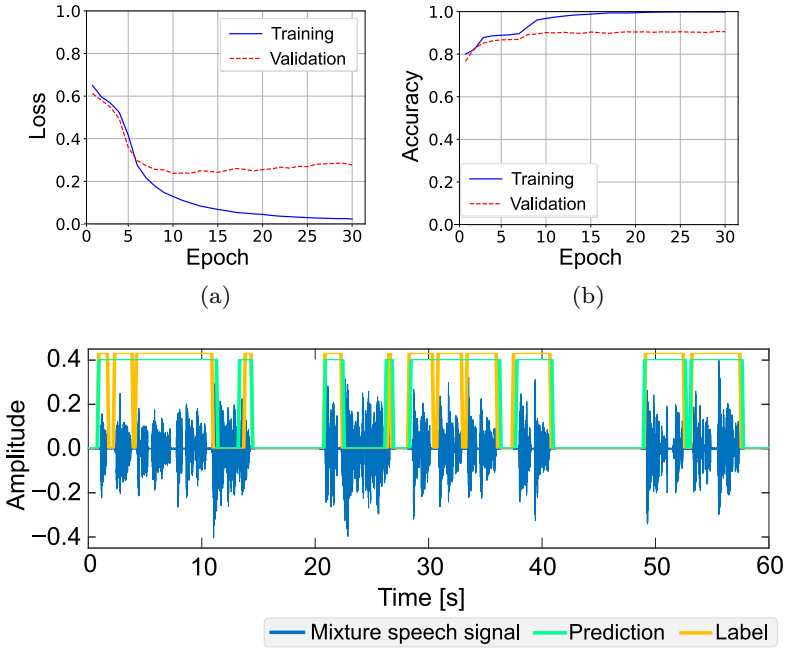
Fig. A.3. Experimental result when $\vartheta = 7.324\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

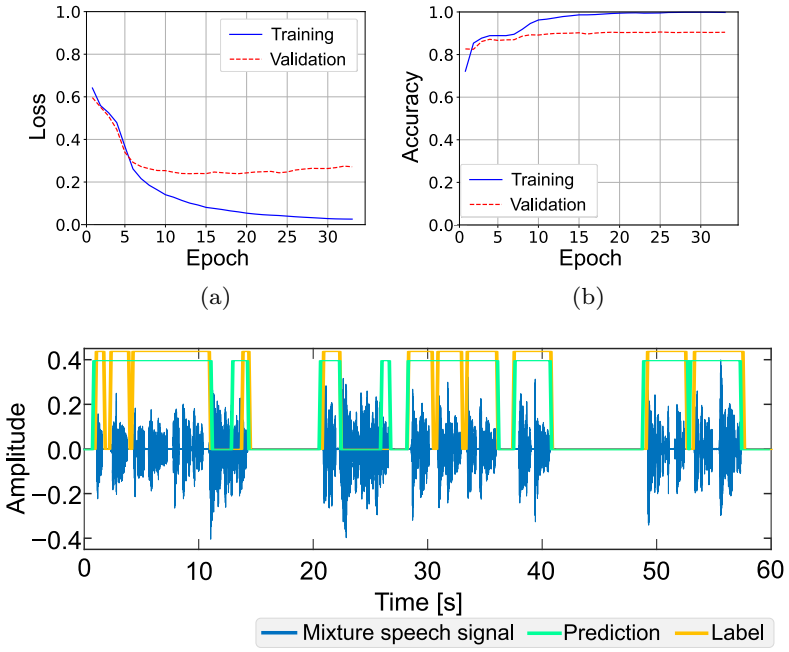
Fig. A.4. Experimental result when $\vartheta = 9.766\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

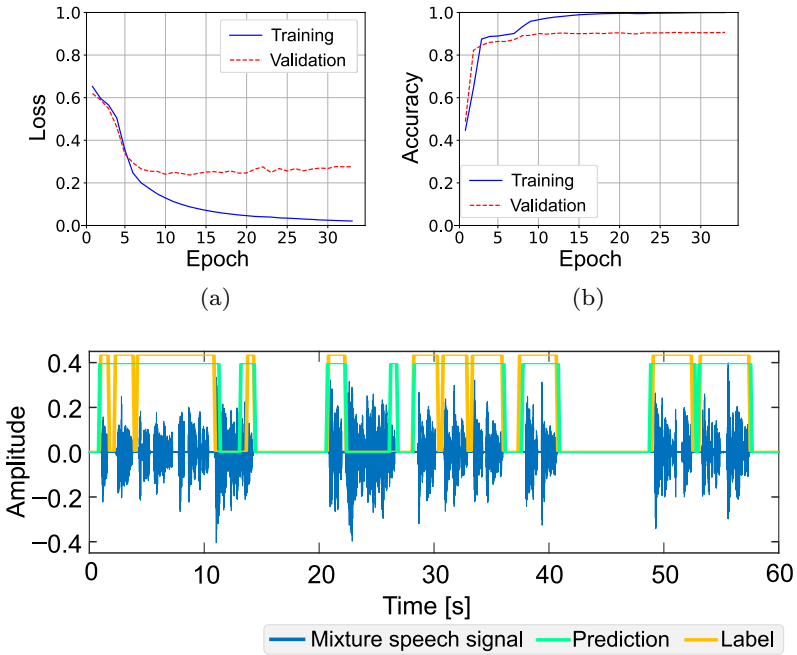
Fig. A.5. Experimental result when $\vartheta = 12.207\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

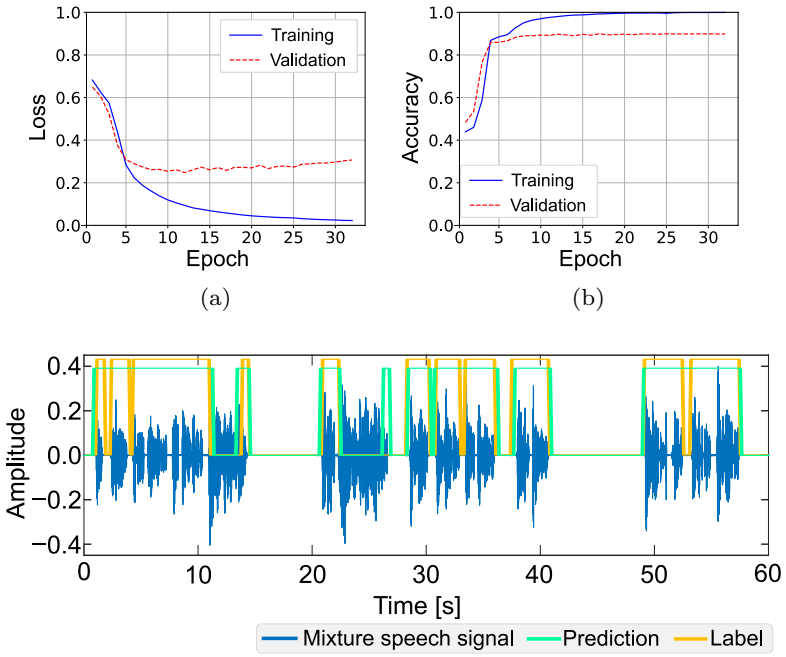
Fig. A.6. Experimental result when $\vartheta = 14.648\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

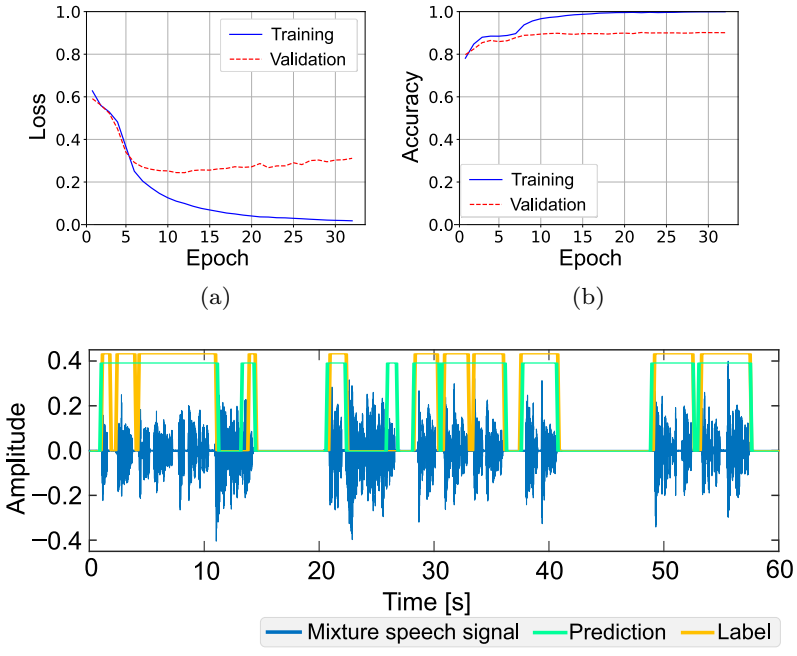
Fig. A.7. Experimental result when $\vartheta = 17.090\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

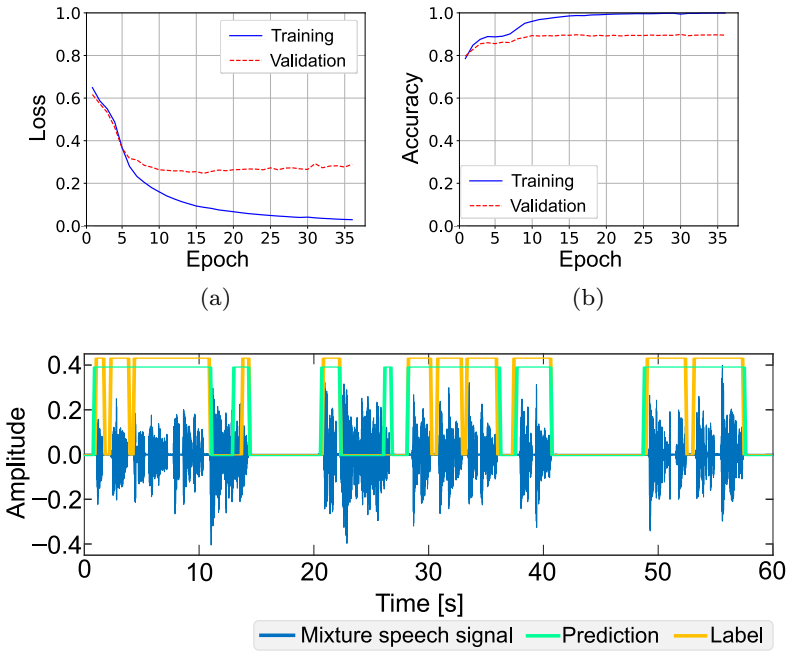
Fig. A.8. Experimental result when $\vartheta = 19.531\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

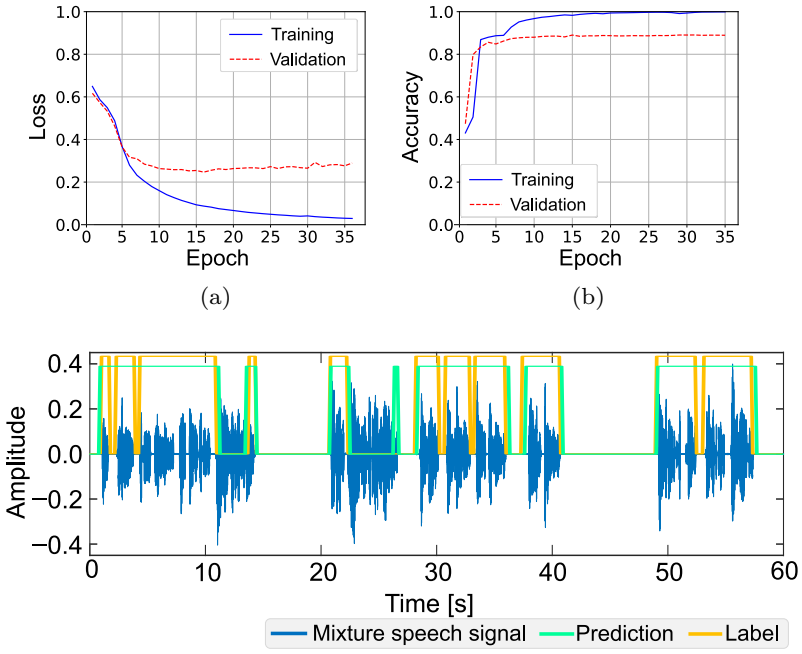
Fig. A.9. Experimental result when $\vartheta = 21.973\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

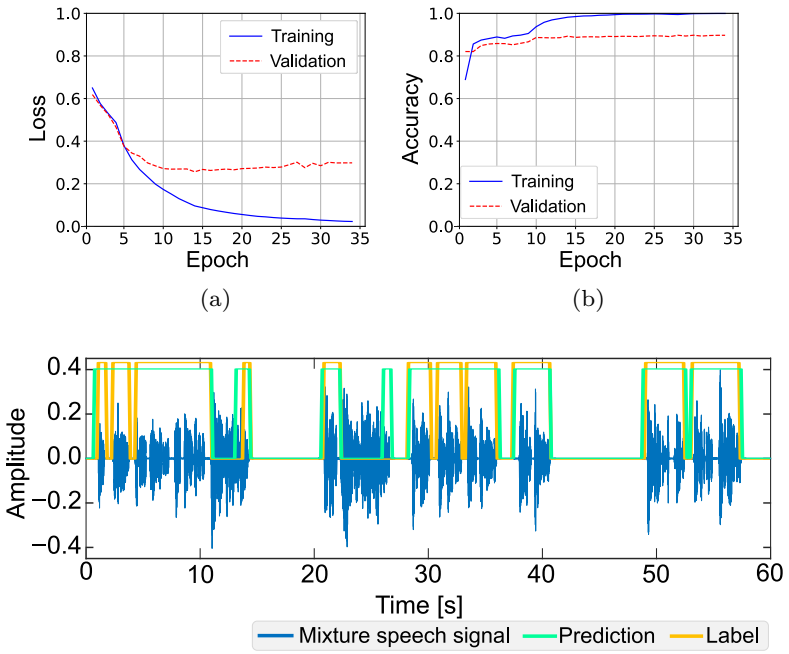
Fig. A.10. Experimental result when $\vartheta = 24.414\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

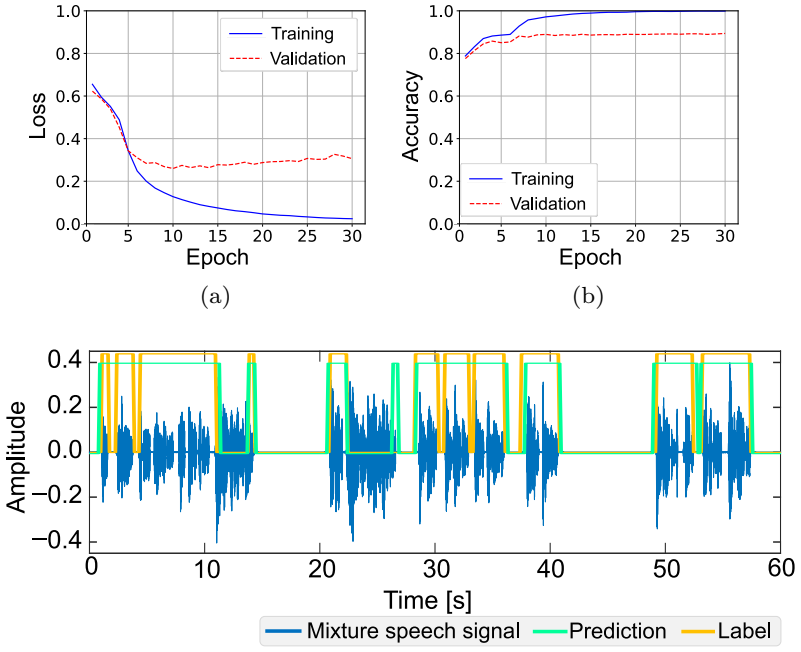
Fig. A.11. Experimental result when $\vartheta = 26.855\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

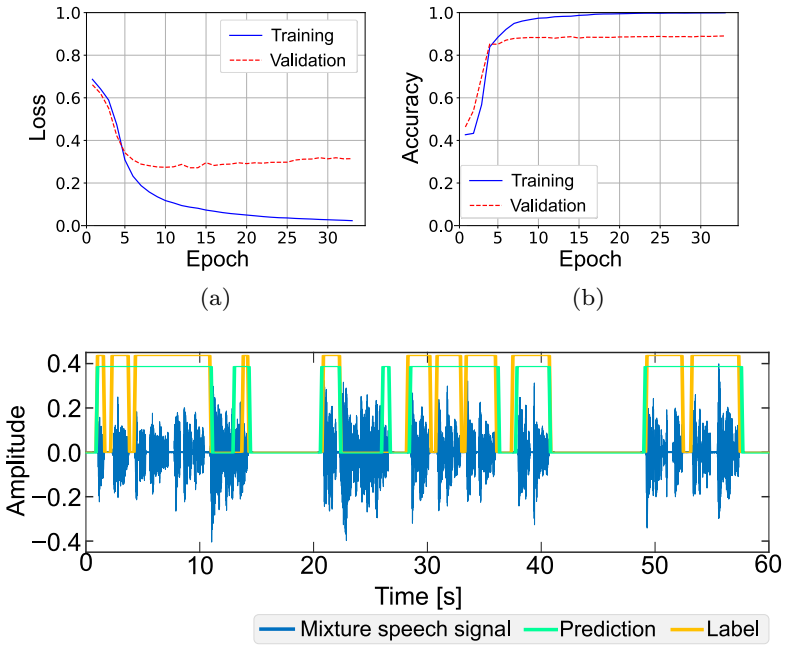
Fig. A.12. Experimental result when $\vartheta = 29.297\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

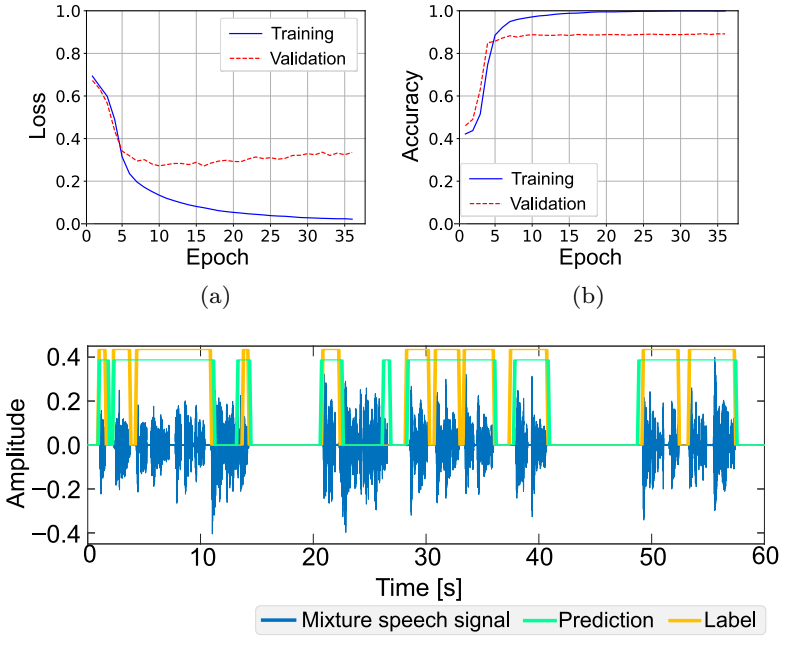
Fig. A.13. Experimental result when $\vartheta = 31.738\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

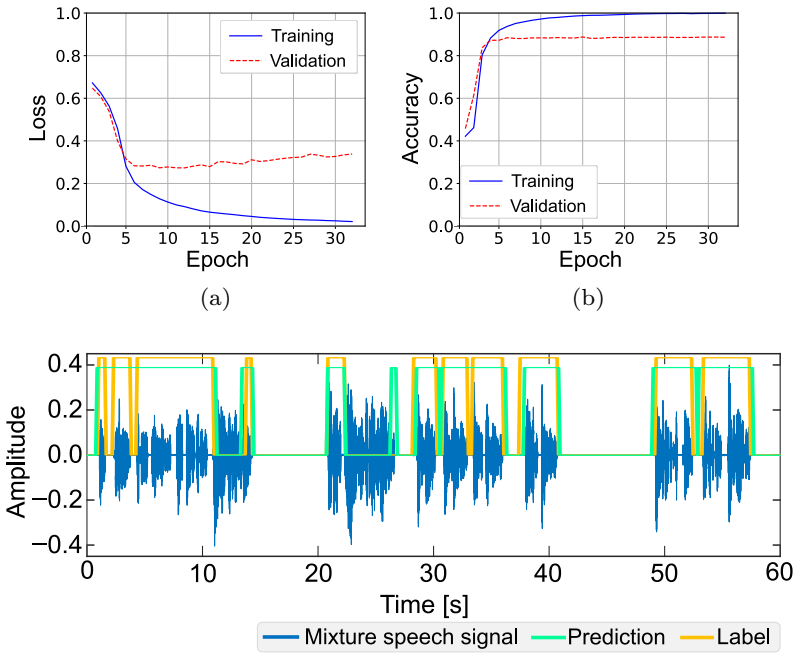
Fig. A.14. Experimental result when $\vartheta = 34.180\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

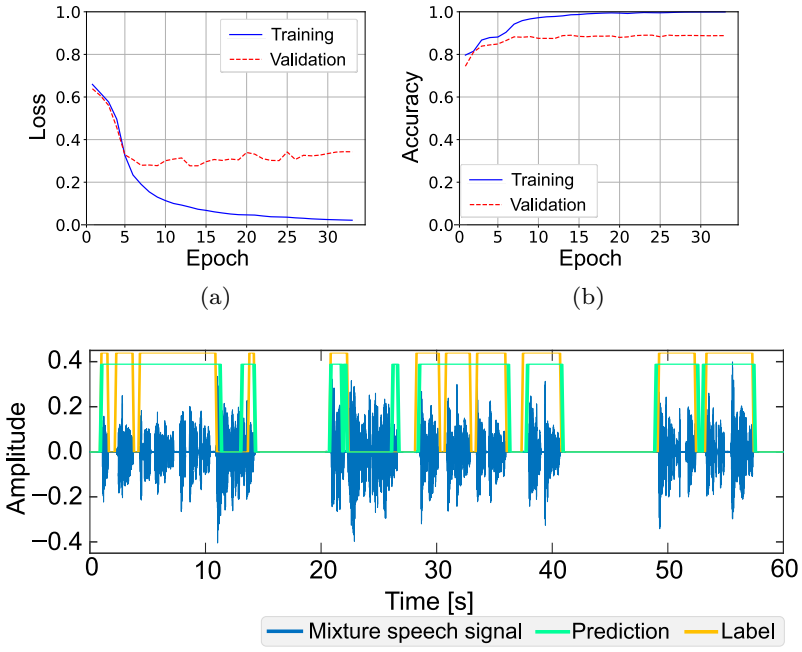
Fig. A.15. Experimental result when $\vartheta = 36.621\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

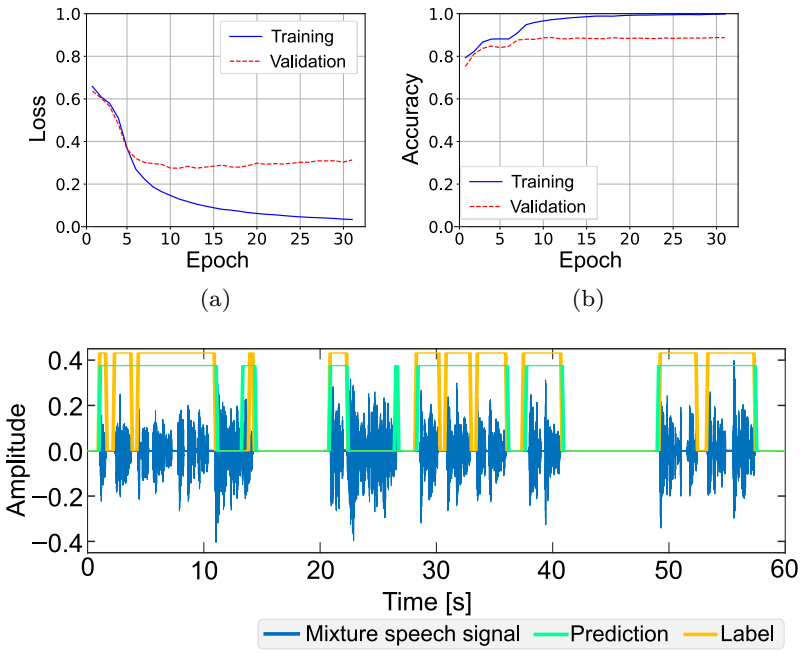
Fig. A.16. Experimental result when $\vartheta = 39.063\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

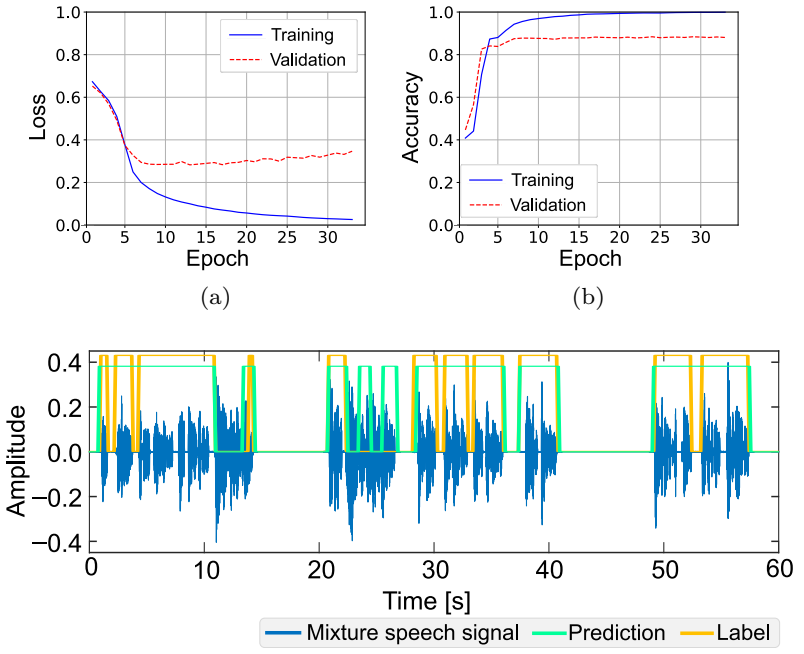
Fig. A.17. Experimental result when $\vartheta = 41.504\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

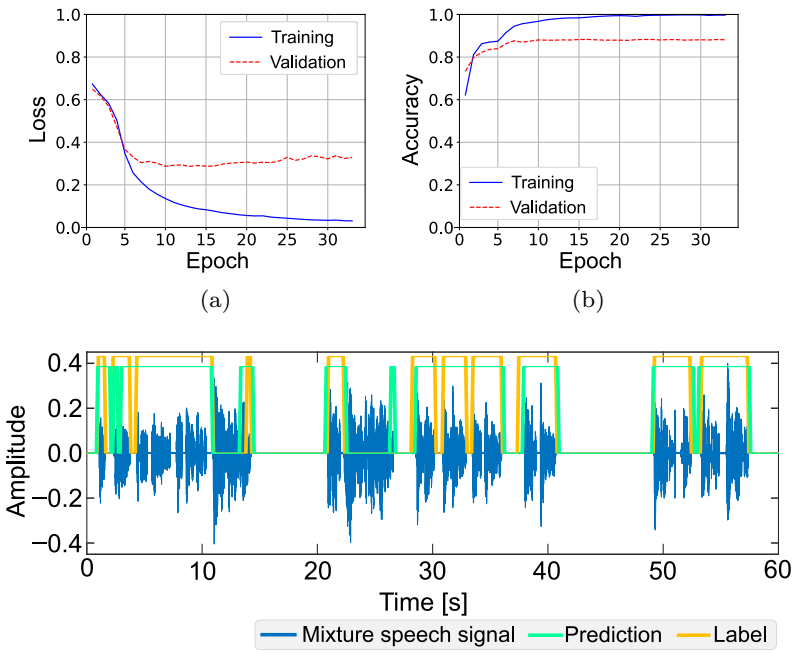
Fig. A.18. Experimental result when $\vartheta = 43.945\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

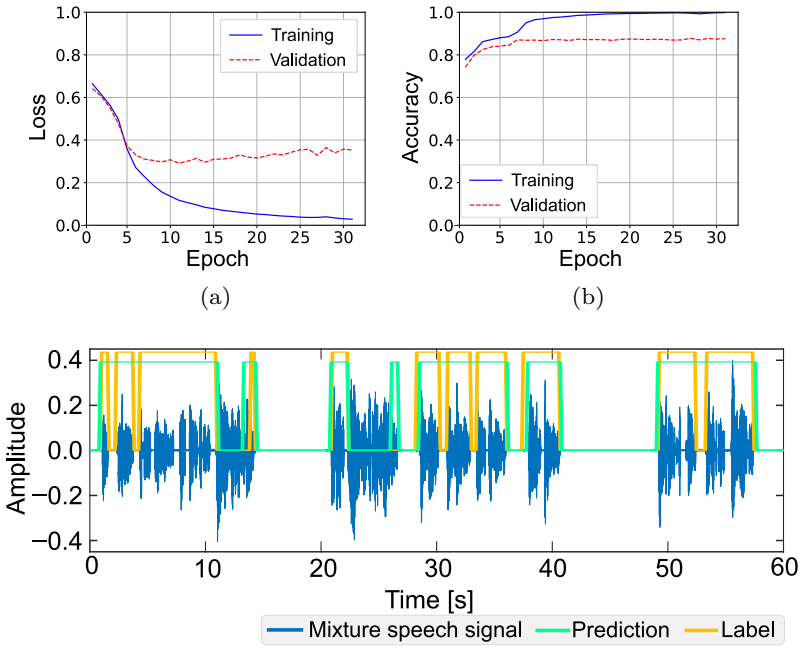
Fig. A.19. Experimental result when $\vartheta = 46.387\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

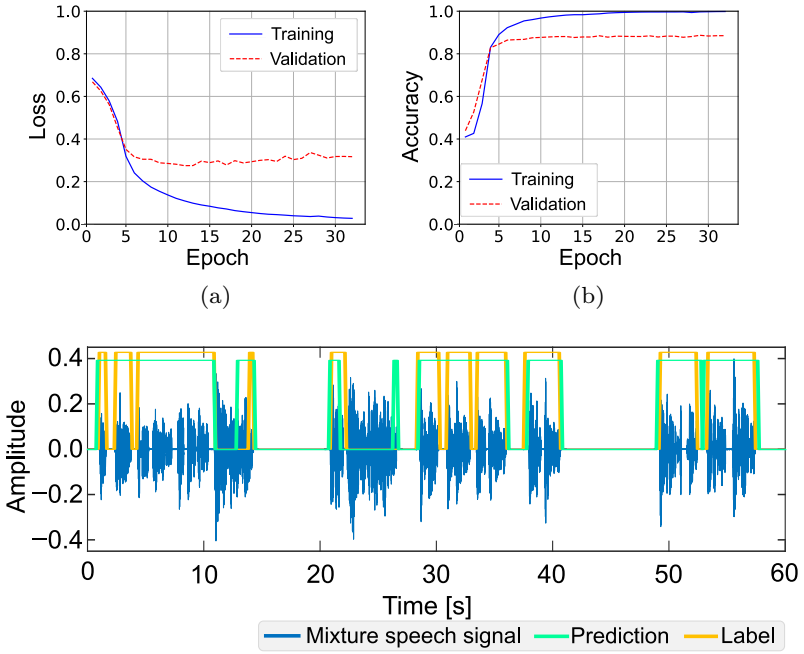
Fig. A.20. Experimental result when $\vartheta = 48.828\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

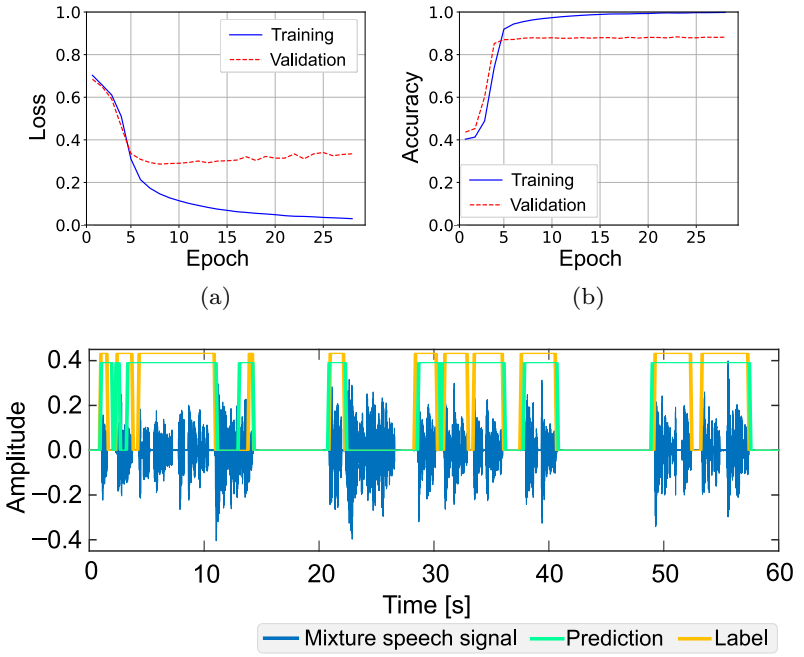
Fig. A.21. Experimental result when $\vartheta = 51.270\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

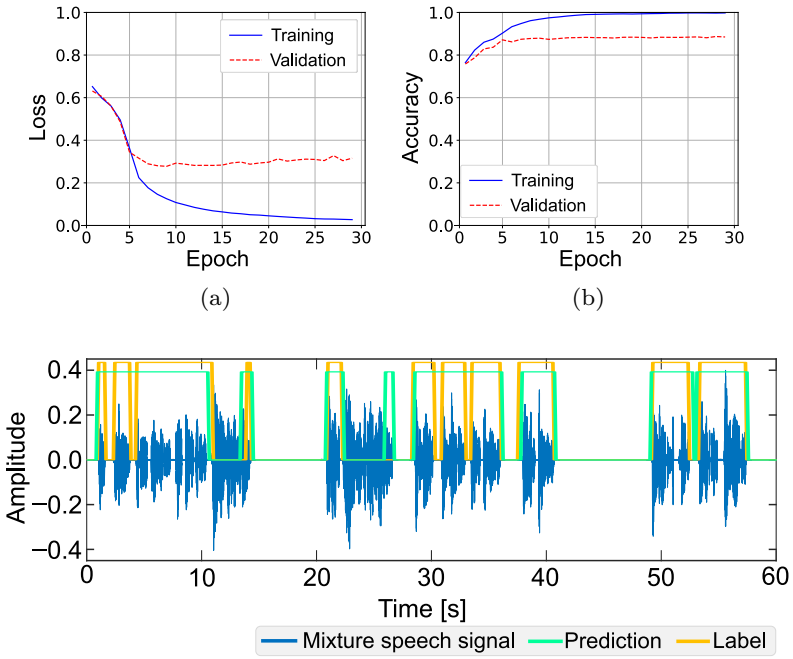
Fig. A.22. Experimental result when $\vartheta = 53.711\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

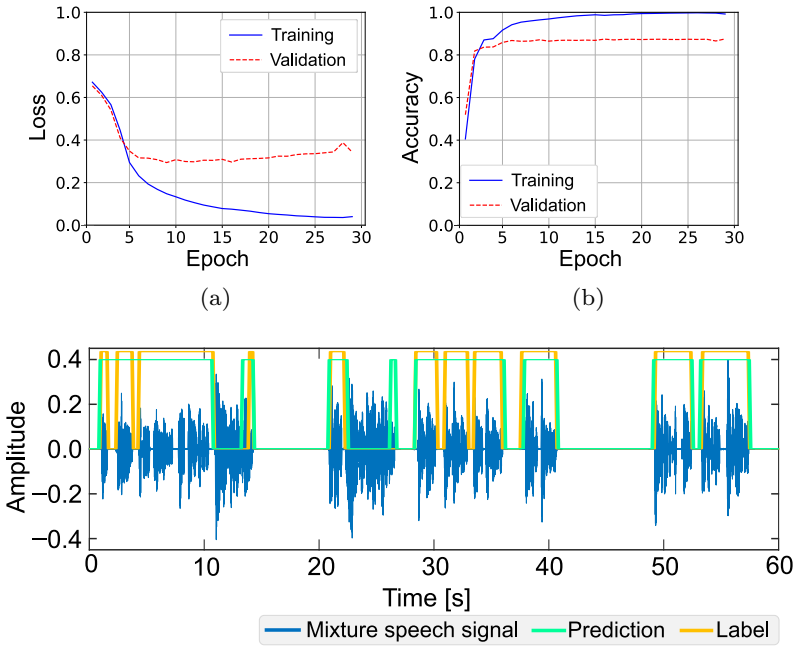
Fig. A.23. Experimental result when $\vartheta = 56.152\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

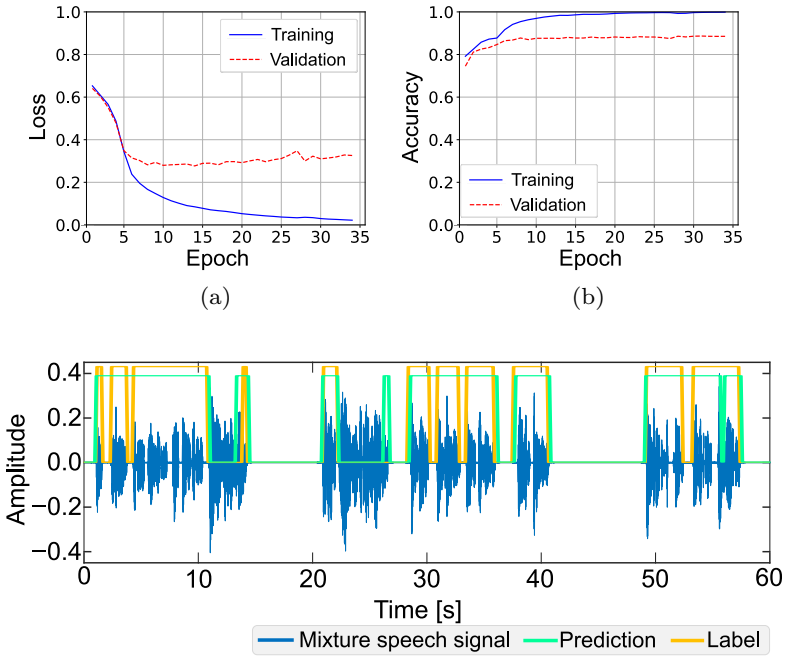
Fig. A.24. Experimental result when $\vartheta = 58.594\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

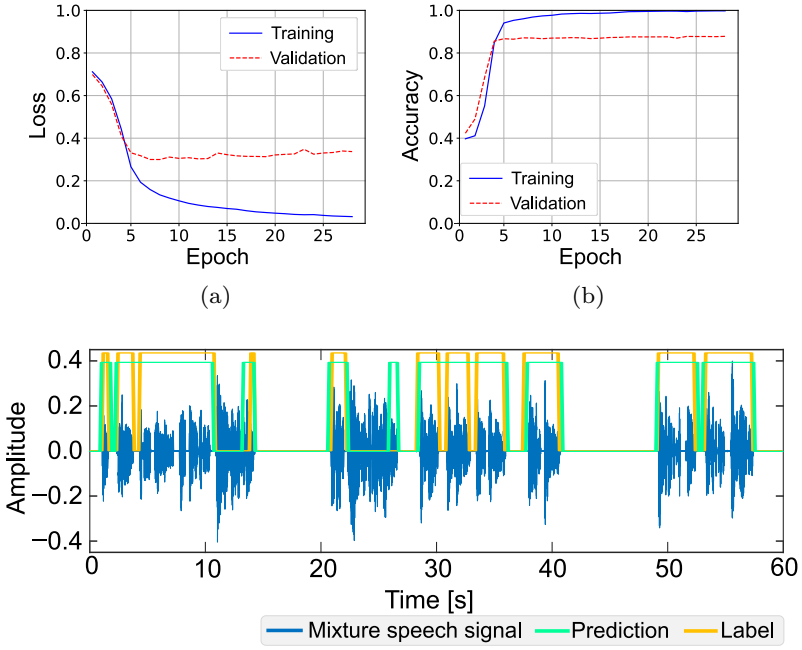
Fig. A.25. Experimental result when $\vartheta = 61.035\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

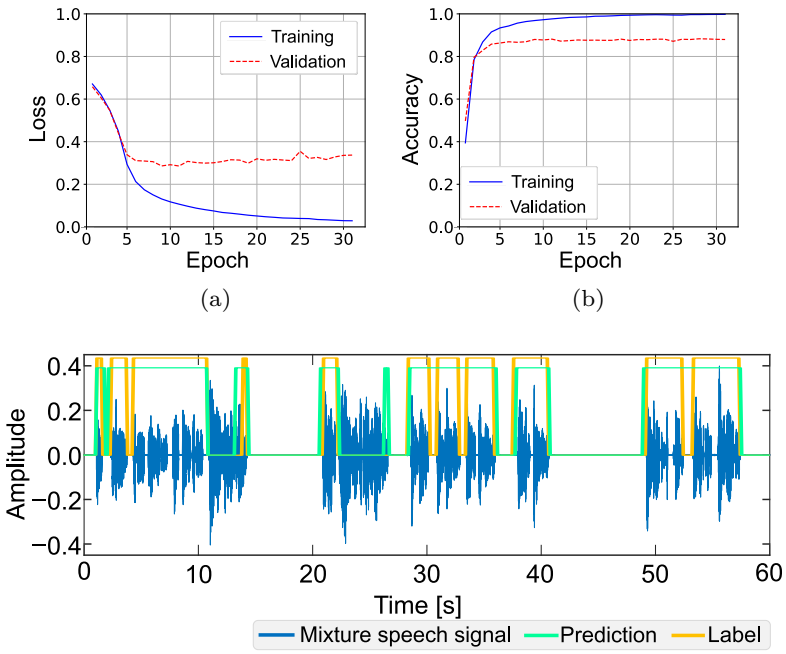
Fig. A.26. Experimental result when $\vartheta = 63.477\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

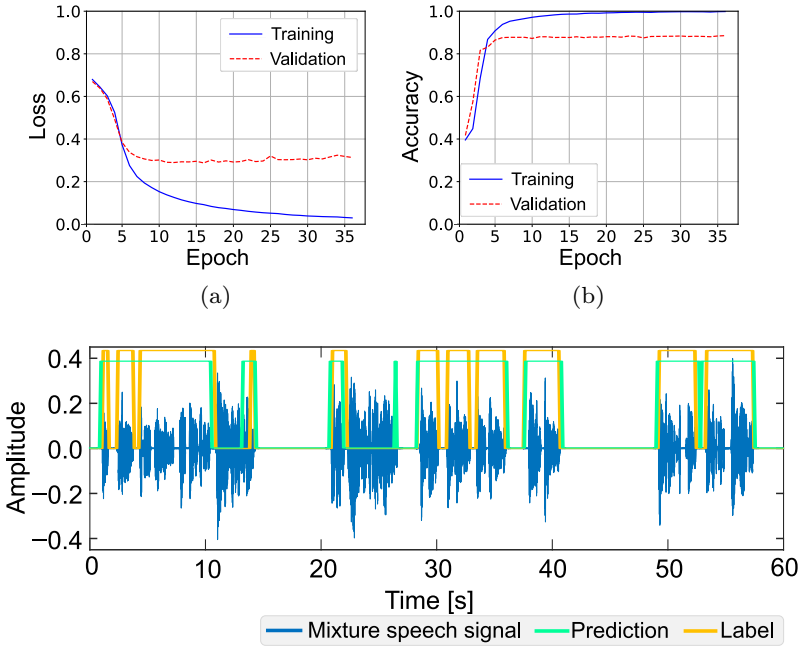
Fig. A.27. Experimental result when $\vartheta = 65.918\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

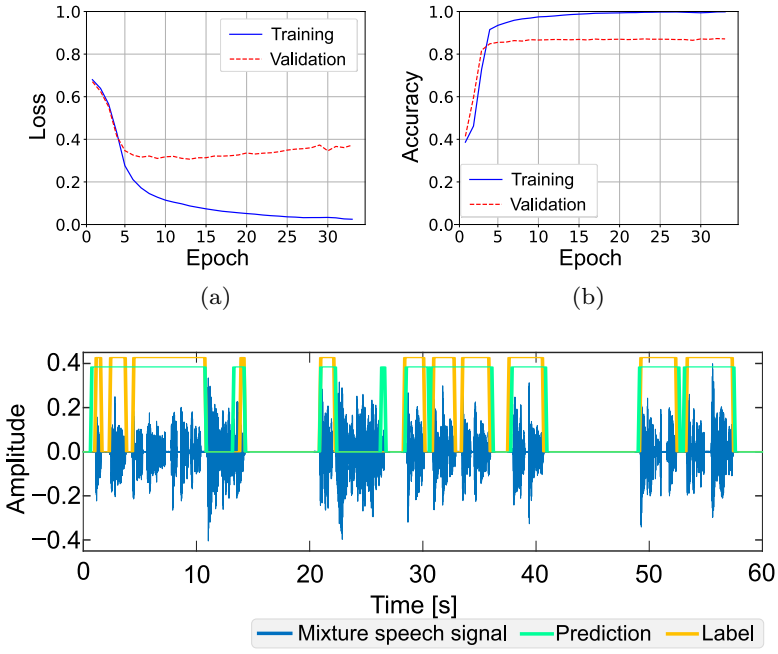
Fig. A.28. Experimental result when $\vartheta = 68.359\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

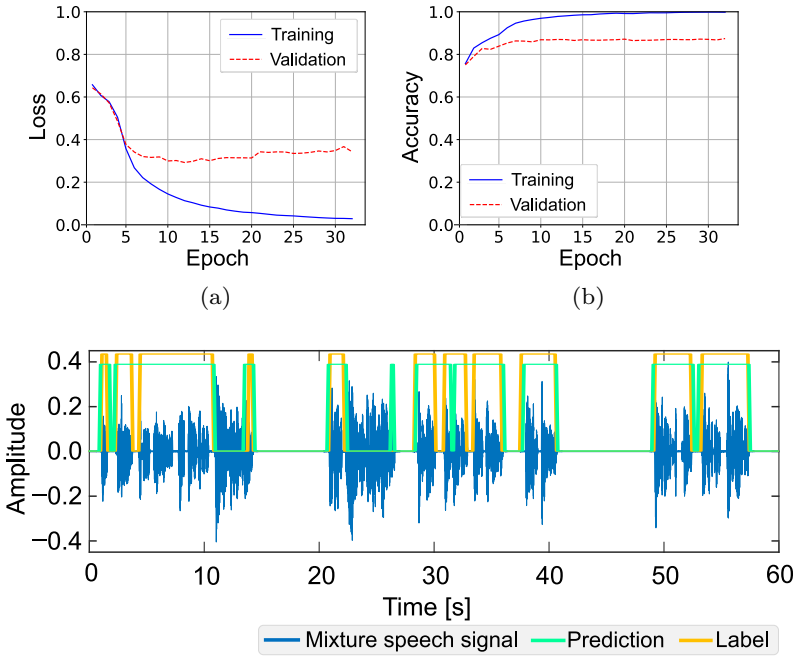
Fig. A.29. Experimental result when $\vartheta = 70.801\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

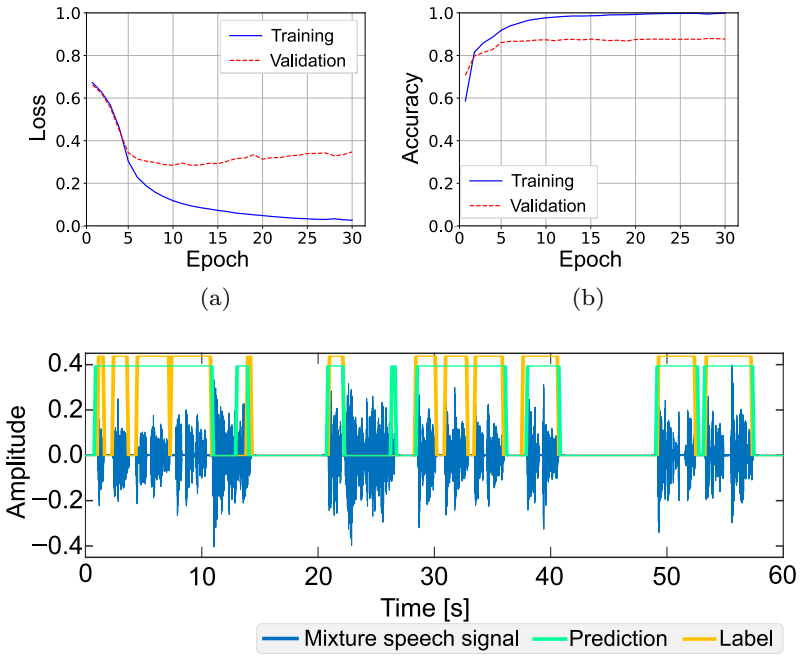
Fig. A.30. Experimental result when $\vartheta = 73.242\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

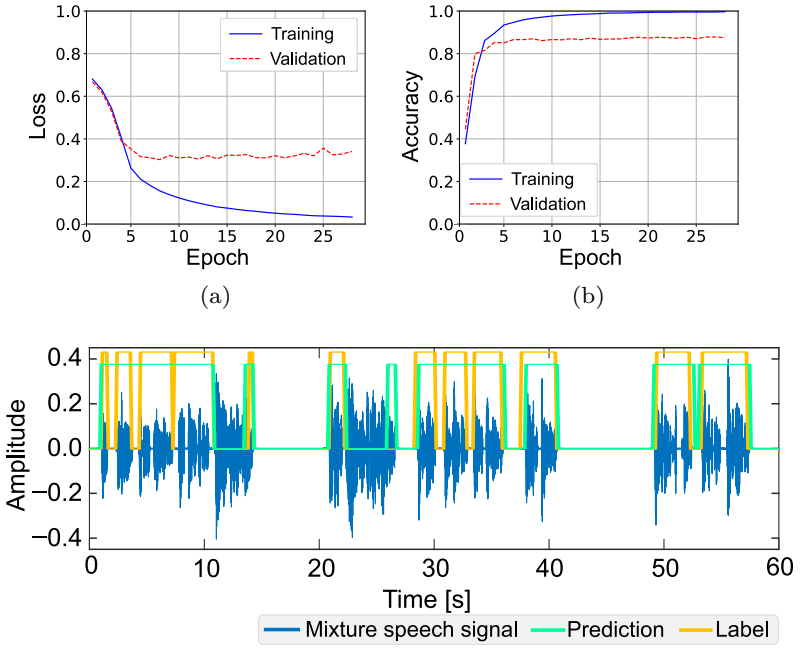
Fig. A.31. Experimental result when $\vartheta = 75.684\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

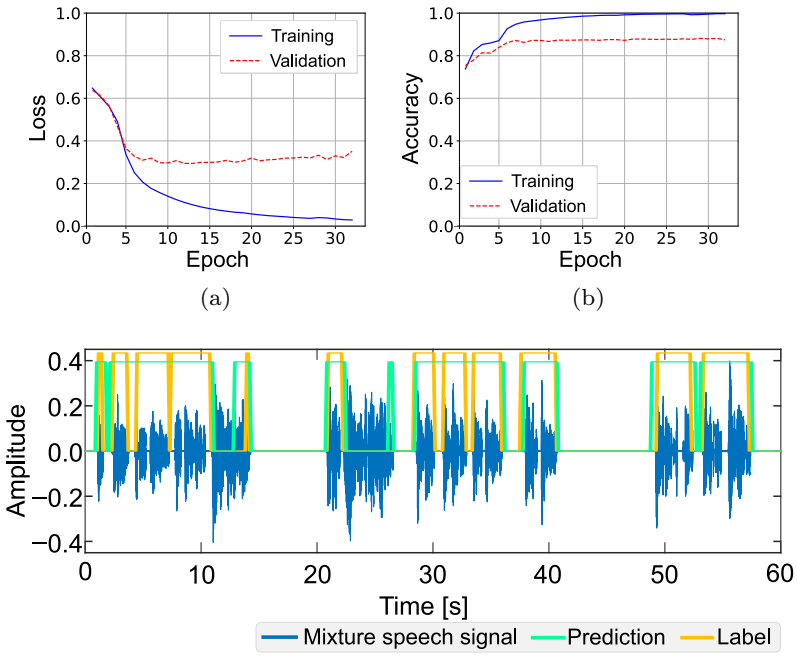
Fig. A.32. Experimental result when $\vartheta = 78.125\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

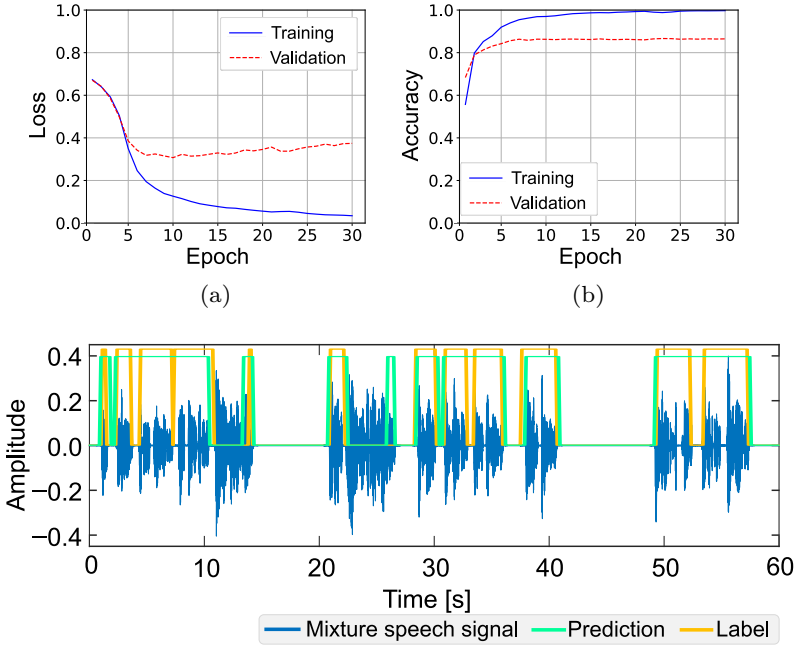
Fig. A.33. Experimental result when $\vartheta = 80.566\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

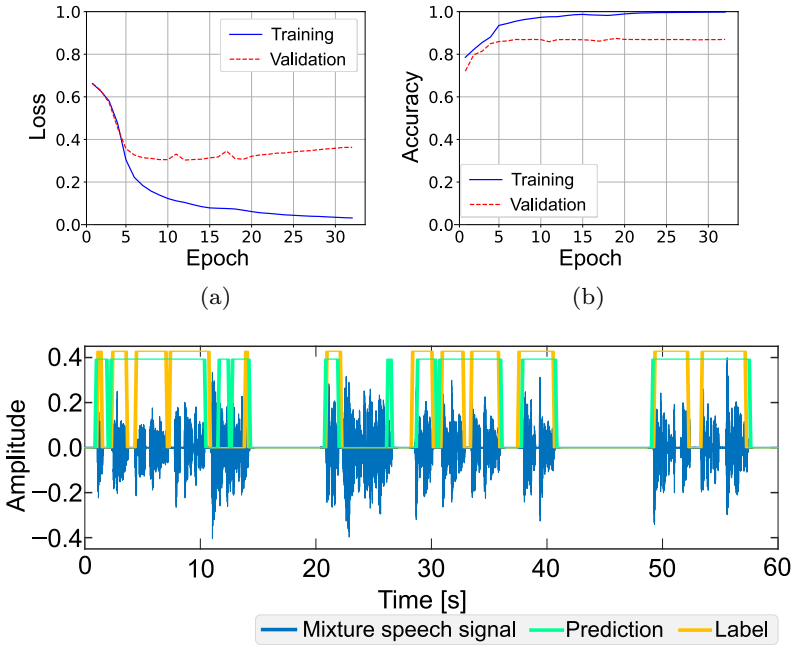
Fig. A.34. Experimental result when $\vartheta = 83.008\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

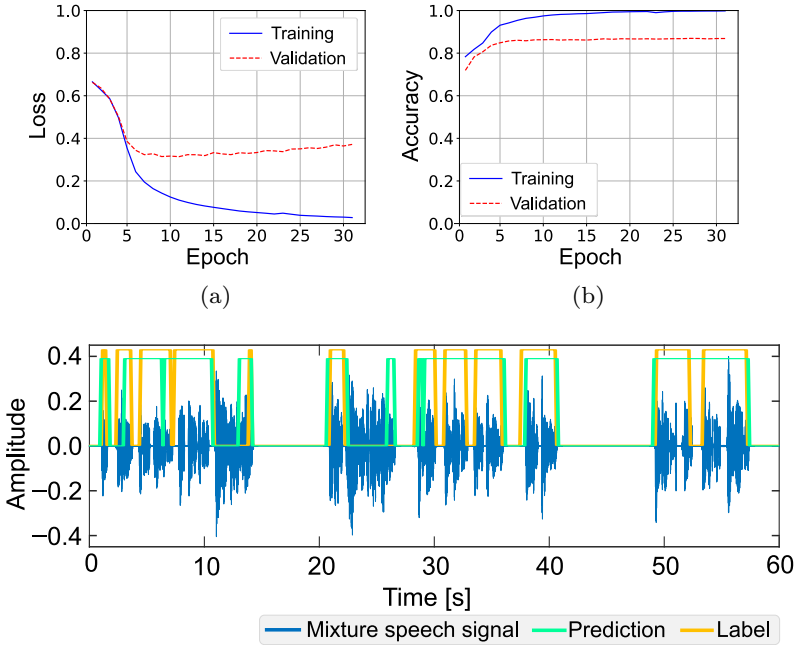
Fig. A.35. Experimental result when $\vartheta = 85.449\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

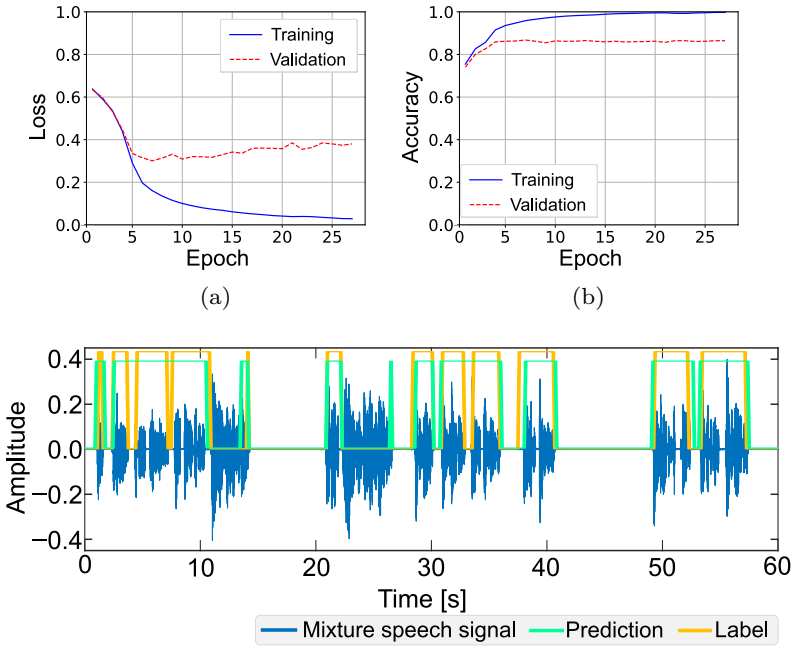
Fig. A.36. Experimental result when $\vartheta = 87.891\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

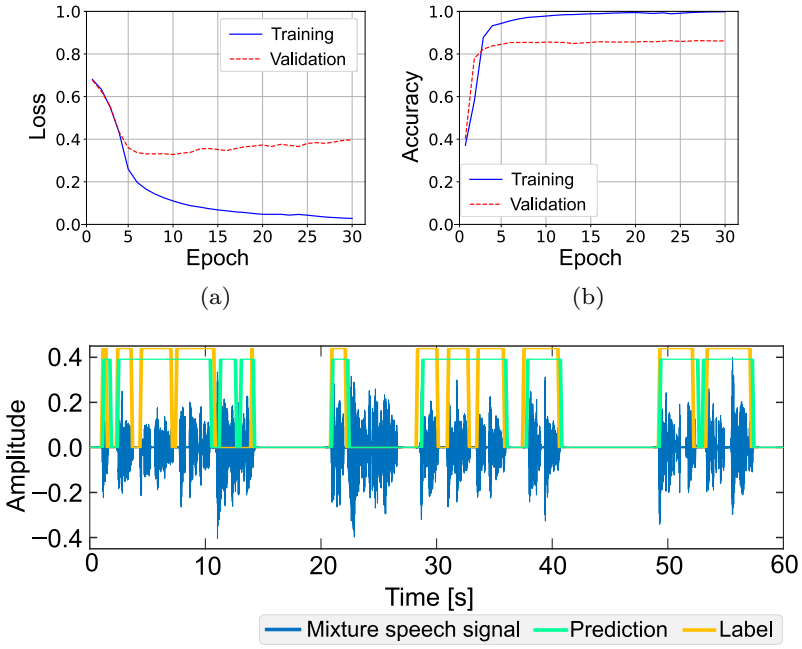
Fig. A.37. Experimental result when $\vartheta = 90.332\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

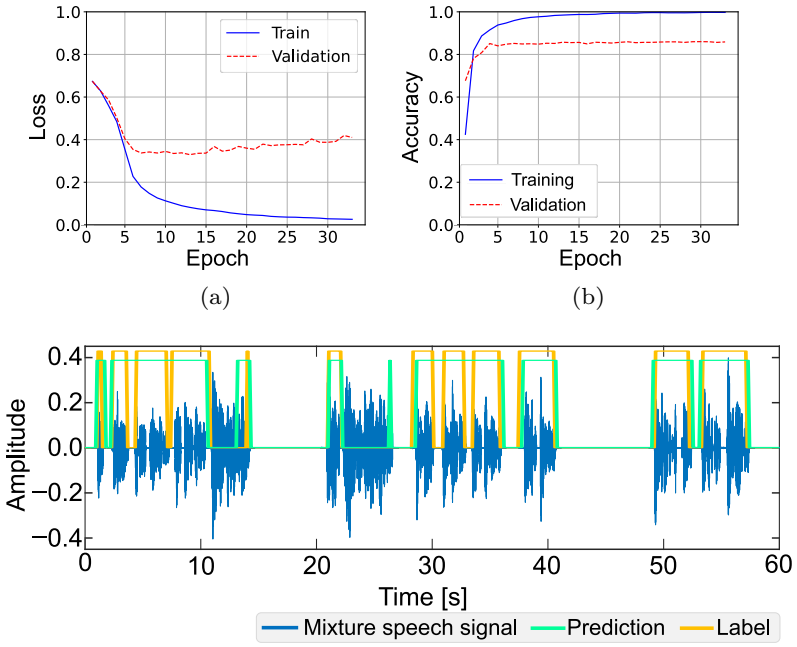
Fig. A.38. Experimental result when $\vartheta = 92.773\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a) (b)

(c)

Fig. A.39. Experimental result when $\vartheta = 95.215\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.



(a)　(b)

(c)

Fig. A.40. Experimental result when $\vartheta = 97.656\%$: (a) loss, (b) accuracy behaviors, and (c) example of prediction and label, respectively.