修士論文

敵対的学習を適用した End-to-end音声認識

2018 年 2 月 1 日

指導教員　峯松 信明 教授

東京大学大学院 工学系研究科
電気系工学専攻

37-166502　増田 嵩志

内容梗概

音声認識技術は近年急速な進歩を遂げており, 音声書き起こしだけでなく音声検索, 音声対話, 音声翻訳など現在では多くの実用化がなされている. この性能向上はニューラルネットワークの導入による影響が大きく, 近年深層ニューラルネットワークに基づくモデルの研究が盛んに行われている. 特に End-to-end 音声認識は音声特徴量から文字を直接出力するモデルで, 所望の誤差基準に近い基準でネットワーク全体が最適化されるだけでなく, デコード機構も非常にシンプルであり, 従来の音声認識の枠組みである DNN-HMM ハイブリッド方式と同等以上の精度を達成している. しかし単一のコーパスのみで学習を行う End-to-end 音声認識はコーパスの性質を受けやすく, 汎化性能に留意する必要がある. 特に従来の音声認識においても長らく課題とされている耐雑音性については, End-to-end 音声認識の枠組みではほとんど扱われていないのが現状である.

そこで本研究では, 耐雑音性を含めた汎化性能をより向上させるための手法として敵対的学習の枠組みを End-to-end 音声認識に適用させることを提案した. 深層学習を用いた認識器は高い識別精度を達成しているがその学習過程の性質により, 正しい識別が可能なサンプルに対して, モデルが誤識別するような小さなノイズを加えて生成された敵対的サンプルを作成することが可能である. このようにして生成された敵対的サンプルをも対象に学習を行なうのが敵対的学習であり, 汎化性能を向上させることができる手法である. この時ノイズを加える枠組みは音声に雑音を付与する処理とみなすことができ, この手法によって付与される雑音は種類を全く仮定する必要がなく, 学習の各ステップで毎度自動で算出されるため, 耐雑音性の向上につながることが期待される. また学習データの明示的な増量が不要であるため, 学習データの増量による学習時間の増加, 最適化の難化といった問題点を回避することが出来る.

TIMIT を対象とした小規模音素認識実験では静音環境下および雑音環境下ともに精度改善が確認された. 一方で WSJ0 や Aurora4 といった 中規模音声認識実験では, それぞれのコーパスでの学習では精度向上が確認でき汎化性能を向上させることにいたったが, 学習コーパスに雑音を付与する方法と比較すると大きく精度が劣っており, 単に敵対的学習を適用させた場合には耐雑音性の向上においては有効でないことが判明した. 敵対的学習の枠組みで生成したノイズが付与された特徴量は, ノイズ付与前に対してほとんど差が見られず, 音声認識における現実の雑音付与の場合と大きく異なる結果が得られた.

以上を踏まえ, ノイズの加え方に課題があると考え, 話者の声道長変換を考慮した場合のアフィン変換による特徴量の変換によって敵対的サンプルを生成し, 音声認識実験でその影響を観測した. 学習コーパスに雑音を付与する方法には依然として耐雑音性の面で及ばないが, ノイズの加え方を修正したことによってエラー率の改善が見られた. 今回導入した敵対的学習の枠組みをノイズの探索空間およびノイズの加え方の面で更に拡張することで, 耐雑音性の向上が達成可能だと考えられる.

目次

図目次

表目次

第1章

序論

1.1　本研究の背景

アルゴリズムの確立や計算機ハードウェアの性能改善によって音声認識の技術が飛躍的な性能改善を遂げている．現在では様々な実用化が行われており, スマートフォンでは Siri[1] やしゃべってコンシェル[2] といった音声検索やアシスタントアプリが搭載され, 多くの人に認知されている．また 2017 年には, Google Home[3] や Amazon Echo[4] のようなスマートスピーカーと呼ばれる, 無線通信接続機能と音声操作のアシスタント機能を持つスピーカーが発売され, 世間の注目を集めている．音声認識の需要はより一層増しており, 深層ニューラルネットワークに基づくモデルにより更なる性能の向上が実現されている．

特に近年では, 2014 年に音声特徴量から文字を直接出力する End-to-end 音声認識モデルが初めて提案 [1] されてから, End-to-end 音声認識に関する研究が盛んに行なわれている．音声認識はある音声データが与えられた際にそれに対応する単語列を推定するタスクであり, 従来の音声認識システムは, 音韻的な最初単位である音素と音響特徴量を結びつける音響モデル, 単語列が言葉として妥当であるかどうかを判定する言語モデルをはじめとする複数のモデルによって構成され, 複雑な工程を経てデコードが行なわれることによって音声認識が実現されていた．この時, それぞれの構成要素は各々の学習基準によってそれぞれ最適化が行なわれる [2]．それに対し End-to-end モデルは基本的に単一のニューラルネットワークのみで構成され, 所望の誤差基準に近い基準でネットワーク全体が最適化されるだけでなく, デコード機構も従来のシステムと比べて単純である．End-to-end 音声認識には, Connectionist Temporal Classification [3] に基づく手法と, Attention mechanism [4] に基づく手法の大きく分けて 2 つの手法が提案されており, 現在盛んに研究が行われている．

しかしながら, 構成要素を別々のコーパスで学習させていた従来の音声認識システムとは異なり, End-to-end 音声認識では一つのコーパスを元に学習が行われるため, コーパスの性質を受けやすい傾向にある．例えば clean な音声で構成されているコーパスを使って学習させた場合, 当然のように耐雑音性は著しく低くなる．したがって End-to-end モデルを構築する場合は汎化性能に留意する必要があると言える．耐雑音性をはじめとした汎化性能を向上させる手法として, 大量の音声データおよび書き起こし文を用意してネットワークを学習させるというデータドリブンな手法が考えられるが, この手法は学習に非常に長い時間を要するだけでなく, 言語によっては十分な量の学習データが手に入らない, 等の問題点がある．耐雑音性は音声認識の抱える問題の一つ [5] であり, 特に End-to-end 音声認識ではあまり議論されておらず, 実際に雑音環境下での認識率は静音環境下の認識率と比べて劣悪である [6, 7] ことからも汎化性能が不十分であるといえる．Amodei ら [7] は, 耐雑音性の向上を目的として学習データに雑音を付与することで学習データの水増しを行なっているが, 前述のような学習時間の問題があるだけでなく, 雑音が混入した音声データを用いることでネットワークの最適化を難化させてしまう可能性が高い．また, この手法によって特定の種類の雑音に対して耐雑音性を向上させることが出来ても未知の雑音に対する性能は保証されないという問題が残る．

[1] https://www.apple.com/jp/ios/siri/
[2] https://www.nttdocomo.co.jp/service/shabette_concier/
[3] https://store.google.com/product/google_home?hl=ja
[4] https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E

1.2　本研究の目的

本研究では, 耐雑音性を汎化性の一つとして捉え, 敵対的学習の枠組み [8,9] を End-to-end 音声認識に応用することを提案する. 深層学習を用いた認識器は驚異的な精度を達成しているが, モデルの性質により, 正しい識別が可能なサンプルに対して, モデルが誤識別するような小さな摂動を加えて生成した敵対的サンプル [10] を作成することが可能である. このようにして生成された敵対的サンプルをも対象に学習を行なうのが敵対的学習であり, End-to-end 音声認識の汎化性能を向上させることが可能であると考えられる. また, 摂動を加える枠組みは音声に雑音を付与する処理とみなすことができ, この手法によって付与される雑音は種類を全く仮定する必要がなく, 学習の各ステップで毎度自動で算出されるため, 耐雑音性も向上させることが可能であると考えられる. このような枠組みでは学習データの明示的な増量が不要であるため, 前述した問題点を回避することが出来る.

1.3　本論文の構成

本論文の構成は以下の通りである.

第 1 章: 序論
本研究の背景と目的について述べる.

第 2 章: End-to-end 音声認識
これまでの音声認識識技術の研究の基礎について述べたのちに, End-to-end 音声認識の既存の枠組みについて説明する.

第 3 章: 敵対的学習
深層学習における敵対的サンプルおよびそれに基づく敵対的学習について述べ, 本研究で扱う敵対的学習の具体的な手法を紹介する.

第 4 章: End-to-end 音声認識における敵対的学習の適用
本研究で提案する, 敵対的学習を適用させた End-to-end 音声認識の枠組みについて説明する. また音素認識実験や音声認識実験を行い手法の評価を行なう.

第 5 章: 声道長変換を考慮した敵対的学習の適用の検討
声道長に対する音響特徴量の依存性について説明を行なったのち, 本研究における課題とされた観点について, 声道長変換を考慮する手法によって検討を行う.

第 6 章: 結論
本研究のまとめについて述べる. また本研究に残された課題と今後の改善案について述べる.

第2章

End-to-end音声認識

2.1　はじめに

本章では, これまでの音声認識識技術の研究の基礎について, 具体的な枠組みの定式化および用いられる特徴量, 評価方法について述べる. その後 End-to-end 音声認識について概説したのち, それを実現する枠組みについて説明を行なう.

2.2　従来の音声認識の枠組み

2.2.1　音声認識の定式化

音声認識は, 入力信号を音響特徴量ベクトルに変換し, その音響特徴量ベクトルの系列から対応する単語列を推定することによって行われる. 音響特徴量は, 入力信号をフレームと呼ばれる一定区間ごとに分析を行ない, 特徴量ベクトルに変換することによって得られる. 単語列の推定では, 音響特徴量ベクトルを時間軸に沿って並べた系列 $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_t, \cdots\}$ から, それに対応する単語列 $W = \{w_1, w_2, \cdots, w_m, \cdots | w_m \in \mathcal{W}\}$ を推定する. なお $\mathcal{W}$ は登場し得るすべての単語の集合である. 入力特徴量ベクトル系列 X が与えられた上での最も出現確率の高い単語列 $\hat{W}$ は, ベイズの定理により以下のように表すことが可能である.

$$\begin{aligned} \hat{W} &= \underset{W}{\operatorname{argmax}}\, p(W|X) && (2.1)\\ &= \underset{W}{\operatorname{argmax}}\, \frac{p(X|W)p(W)}{p(X)} && (2.2)\\ &= \underset{W}{\operatorname{argmax}}\, p(X|W)p(W). && (2.3) \end{aligned}$$

式 (2.3) における第 1 項 $p(X|W)$ は単語列が既知の際の特徴量ベクトル系列の確率密度を表現するモデルであり, 音響モデルと呼ばれる. 一方第 2 項 $p(W)$ は単語列がどのような確率で出現するか, すなわち “単語列としてのもっともらしさ” を記述したモデルであり, 言語モデルと呼ばれる. これらのモデルの出力がデコーダによって統合されることで音声認識が実現されている.

典型的な音声認識システムにおける音響モデルには隠れマルコフモデル (Hidden Markov Model; HMM) という確率過程が用いられ, 以下のように表される [11].

$$\begin{aligned} p(X|W) &= \sum_{\boldsymbol{s}} p(X|\boldsymbol{s})p(\boldsymbol{s}|W) && (2.4)\\ &= \sum_{\boldsymbol{s},\boldsymbol{m}} \left(\prod_t p(\boldsymbol{x}_t|s_t) \right) p(\boldsymbol{s}|\boldsymbol{m})p(\boldsymbol{m}|W). && (2.5) \end{aligned}$$

ここで $\boldsymbol{m}$ は音素系列を表す変数, $\boldsymbol{s}$ は HMM の潜在変数である HMM 状態系列を表す変数である. $p(\boldsymbol{m}|W)$ は音素と単語の関係をモデル化したモデルであり, 発音辞書モデルと呼ばれる. 発音辞書モデルは辞書をベースに人手で定義がなされる.

$p(\boldsymbol{s}|\boldsymbol{m})$ は一般的な HMM と同様, 次はどの HMM 状態に移るかを表現する状態遷移確率を用いて定義される. 一方, $p(\boldsymbol{x}_t|s_t)$ は HMM の出力確率密度関数に相当するモデルであり, 深層学習が浸透する以前は混合正規分布 (Gaussian Mixture Model; GMM) を用いて以下のように定義されることが一般的であった.

$$p(\boldsymbol{x}_t|s_t) = \sum_k \pi_{s_t,k} \mathcal{N}(\boldsymbol{x}_t; \mu_{s_t,k}, \Sigma_{s_t,k}). \quad (2.6)$$

ここで $\mathcal{N}(\boldsymbol{x}_t; \mu_{s_t,k}, \Sigma_{s_t,k})$ は平均ベクトル $\mu_{s_t,k}$, 共分散ベクトル $\Sigma_{s_t,k}$ で示される多変量正規分布の確率密度関数である. 混合正規分布を出力分布とした HMM の最尤推定は EM アルゴリズムの一種である Baum-Welch アルゴリズムで行なわれる.

音声認識の言語モデルとしては, 以下のような N-gram 言語モデルを用いるのが一般的である.

$$p(W) = \prod_m p(w_m|w_{m-1}, w_{m-2}, \cdots, w_1) \tag{2.7}$$

$$\approx \prod_m p(w_m|w_{m-1}, w_{m-2}, \cdots, w_{m-(N-1)}). \tag{2.8}$$

すなわち N-gram 言語モデルでは, ある単語 w_i の出現確率を直前の $N-1$ 個の単語に対するマルコフ過程であると考える. 言語モデルの構築の際には, 訓練データ中に登場する単語列の登場頻度をカウントすることで確率分布を算出するが, 一般の文脈では訓練データ中に登場しない単語列が当然存在する. このようなデータスパースネス問題に対処するため, 様々な平滑化手法と組み合わせて利用されることが一般的である [12].

2.2.2　DNN-HMM ハイブリッド方式による音響モデル

2.2.1 章で述べたように, 音声認識のモデルは大まかに音響モデル・言語モデルの 2 つのモデルに切り分けることができる. 特に, 音響モデルについては, 音響特徴量の分布を GMM でモデル化しその時間的な推移を HMM によってモデル化する GMM-HMM と呼ばれるモデル, 言語モデルについては, N-gram を用いるのが一般的であった. しかしアルゴリズムの発展やハードウェアの性能向上により, 2011 年以降から深層ニューラルネットワーク (Deep Neural Networks; DNN) を用いるモデルが高い精度を達成し, それ以降 DNN を利用するのが一般的となった. 具体的には, 音響モデルについては, 音響特徴量と HMM の隠れ状態の関係を GMM による記述から DNN によるものへと変更した DNN-HMM ハイブリッド方式がスタンダードな方式となった [13]. 一方音響モデルについては, 再帰型ニューラルネットワーク (Recurrent Neural Network; RNN) を用いた RNN 言語モデルを N-gram 言語モデルと併用させた場合に大きな精度改善が見られている [14]. 以下では現在最も広く採用されている DNN-HMM ハイブリッド方式について記述する.

DNN-HMM ハイブリッド方式における DNN は入力音声信号に対応する HMM 状態の確率分布を表現するために用いられる. ここでニューラルネットワークの出力層として softmax 層を用いることで HMM 状態の条件付き確率が表現され, 言語モデル等の他の確率モデルとの統合によってデコードが行なわれる. softmax 出力層の出力ベクトルの k 番目の要素, すなわち k 番目のクラスが出現する確率は以下のように表される.

$$p(k|x) = \frac{\exp(z_k(x))}{\sum_l \exp(z_l(x))}. \tag{2.9}$$

ここで $z_k(x)$ は, ニューラルネットワークにおける k 番目の出力ノードへの入力の総和である. 各ノードの総和が 1 になるように正規化されているため, 出力層を softmax 層にすることで確率分布を表現することが可能になる.

この softmax 出力層を持つニューラルネットワークに音響特徴量 $\boldsymbol{x}_t$ を入力することで, HMM 状態 s_t の事後確率 $p(s_t|\boldsymbol{x}_t)$ を求めることを考える. ここで式 (2.5) における $p(\boldsymbol{x}_t|s_t)$ は, ベイズの法則によって

$$p(\boldsymbol{x}_t|s_t) = \frac{p(s_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t)}{p(s_t)} \tag{2.10}$$

と表せる. $p(\boldsymbol{x}_t)$ は認識を行う時点では定数としてよいので無視できる. 一方 $p(s_t)$ は「音声データに状態 s_t 相当の音がどの程度の頻度で登場するのか」に相当し, あらかじめ GMM を用いて求めることが可能である.

すなわち $p(\boldsymbol{x}_t|s_t)$ を求めるために, GMM を使ってガウス分布の確率値として直接求めるのではなく, DNN の事後確率を用いて, ベイズの法則から間接的に求めるのが DNN-HMM ハイブリッド方式である. GMM-HMM では音響特徴量分布を陽に仮定していたが, この方式ではその仮定が不要なため, より柔軟な分布を表現することが可能である. また GMM-HMM を学習する際には, 統計的推定の信頼性の点から入力特徴量に制約があったが, DNN を利用することでより生の音声データに近い特徴量を入力することが可能になった. これにより識別器に特徴抽出を統合して最適化することが可能になり, 精度の向上に至ったと考えられる [15].

2.2.3　音声認識で用いられる音響特徴量

音声認識ではその精度を向上させるために音声から重要な特徴量を抽出する処理が行われている. 本節では, 音声認識で用いられる数多くの音響特徴量のうち, 深層学習が浸透する以前の音声認識で広く用いられていたメル周波数ケプストラム係数 (Mel-Frequency Cepstrum Coefficient; MFCC), End-to-end 音声認識をはじめとしてそれ以降の音声認識で広く用いられているメルフィルタバンク特徴量の 2 つについて説明を行なう.

まずはじめに, 音声波形に対して高域強調を行ったのち, ハミング窓やハニング窓等の窓関数を乗ずることによって数十 ms 程度のフレームを切り出し, それに対して離散フーリエ変換 (Discrete Fourier Transform; DFT) を施し, パワースペクトルを求める. これは人間が音声を認識する際に位相の情報をほとんど利用せずパワーの情報を主に用いているためである.

次にパワースペクトルに対してメル化と呼ばれる処理を行なう. これは, 人間の聴覚の高周波分解能が低く低周波分解能が高いという性質を反映させたメル尺度に対応させながら, 細かい周波数ビンの値をグループ化する処理である. メル尺度は音高の知覚的尺度であり, より具体的には本来の周波数 f に対して以下の式でメル周波数 $\mathrm{Mel}(f)$ を表すことができる.

$$\mathrm{Mel}(f) = 2595 \log_{10} \left(1 + \frac{f}{700} \right). \tag{2.11}$$

図 2.1 はメルフィルタバンクの模式図であり, フィルタバンクの個数は 20〜40 程度であることが一般的である. 次に, メルフィルタバンクの出力を対数に変換する. これは人間の聴覚の音のパワーに対する分解能が, パワーが大きくなるにつれ小さくなることを反映したものである. 以上の操作によって (対数) メルフィルタバンク特徴量が得られる.

さらに音声のパワースペクトルについて, その微細構造が声門波を, スペクトル包絡成分が声道のインパルス応答を表現しており, 発話内容や話者の声質といった情報は後者に表れるため, これを抽出したケプストラム特徴量が音声認識で広く用いられている. 特に, 対数メルフィルタバンク特徴量に対して離散コサイン変換をかけ, 低い方から通常 10〜15 次元程度の次元を用いたものが MFCC である.

ケプストラム特徴量はスペクトルベースのものになるため, 次元間の相関が低く, GMM の学習に適しており, 深層学習が浸透する以前は MFCC が標準的な音声認識の特徴量とされていた. しかし DNN-HMM ハイブリッド方式のような DNN ベースの音響モデルを学習する際には, 特徴量の次元間の相関は特に問題にならず, むしろより生の音声データに近い特徴量を選択し特徴量抽出まで最適化を行なう方が高い精度が出る [16] ことから, MFCC よりもフィルタバンク特徴量あるいは対数パワースペクトルが使われるようになった.

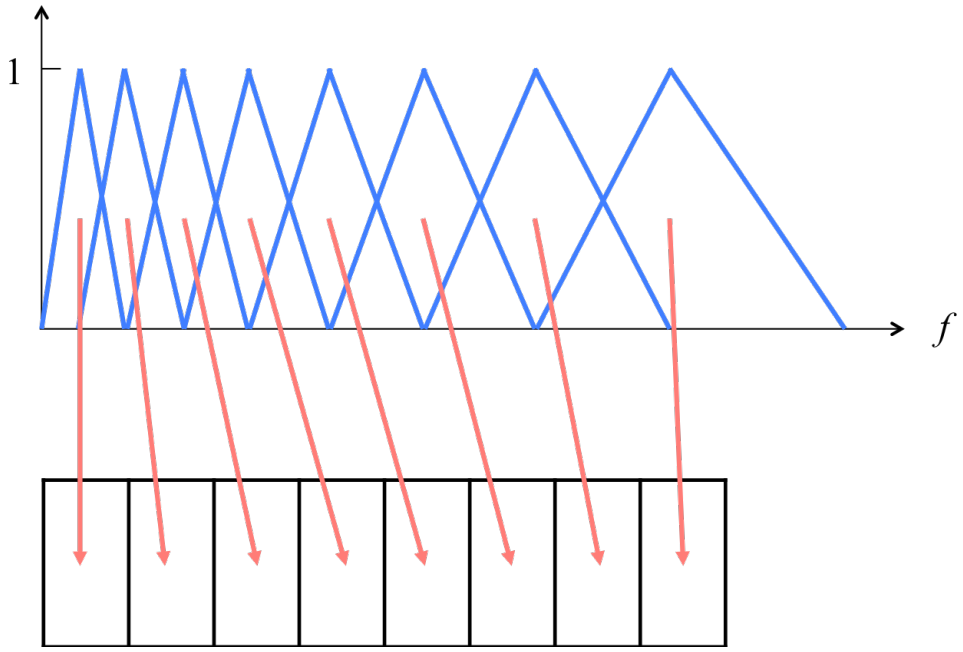


図 2.1: メルフィルタバンク

以上が主要な音響特徴量の求め方についてだが, これらの音響特徴量を用いる際に, 各フレームの静的な特徴だけでなく音声信号の時間的な変化を表す動的な特徴量も有効であり, よく用いられている. 具体的には, あるフレームの特徴量ベクトルの各成分の時間微分であるデルタ (Δ) 特徴量, さらにその時間微分であるデルタデルタ (ΔΔ) 特徴量が一般的によく用いられる.

2.2.4　音声認識の評価方法

音声認識においては, 結果を返すまでの時間が速いほど, また誤識別が少ないほど良い認識器とされている. 前者の応答時間の速さは応用場面では極めて重要な観点であるが, 本研究ではベースライン手法と提案手法とで認識実行時の枠組みが同一なため差がなく評価指標として不適切であるため, 後者についてのみ説明を行なう.

音声認識の認識性能は一般的に単語誤り率 (Word Error Rate; WER) で評価を行なう [17]. 単語単位で認識性能の評価を行なう場合, 誤りは以下の 3 種類に区分することが可能である.

脱落誤り (deletion error)　正解に存在する単語が認識されない

挿入誤り (insertion error)　正解に存在しない単語が誤って挿入される

置換誤り (substitution error)　ある単語を別の単語に間違える

いま, 正解における単語数を N, 脱落誤りが発生している単語数を Del, 挿入誤りが発生している単語数を Ins, 置換誤りが発生している単語数を Sub と表すとすると, WER は以下のように表せる.

$$\mathrm{WER} = \frac{Del + Ins + Sub}{N} \cdot 100\,[\%]. \tag{2.12}$$

すなわち, 動的計画法によって得られる出力と正解の編集距離によって WER は算出され, 値が小さければ小さいほど誤りの少ないシステムであることを表す. ここで WER が必ずしも 0 %〜

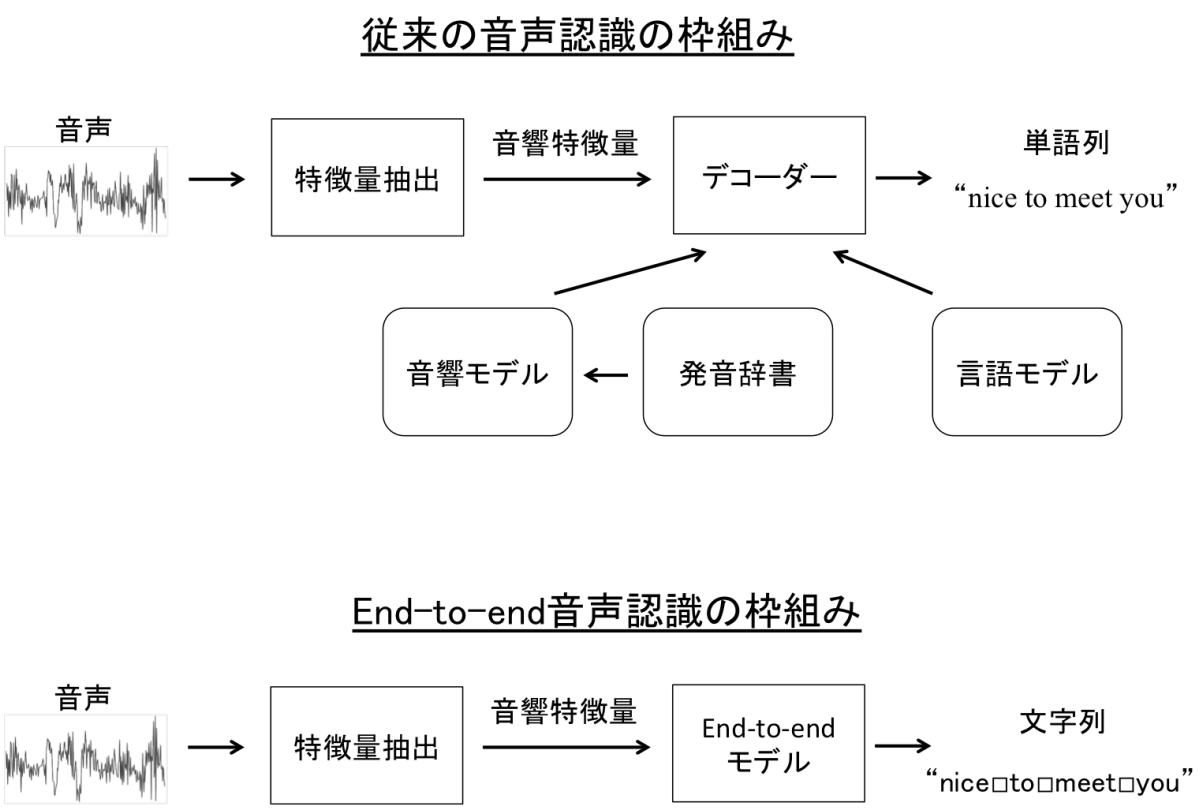


図 2.2: End-to-end 音声認識の枠組みと各モデルの位置づけ

100 % の範囲に収まるわけではないことに注意されたい. また音声認識システムの出力形式が単語単位ではなく文字単位や音素単位である場合には, WER の代わりに文字誤り率 (Character Error Rate; CER) や音素誤り率 (Phoneme Error Rate; PER) で評価を行なう例が多い.

2.3　End-to-end 音声認識の枠組み

2.3.1　概要

従来の音声認識システムは, 音声分析・音響モデル・言語モデル・発音辞書・デコーダーなどの多くの要素から構成されており, それぞれの要素が別々に研究されている. 音声認識の性能向上のためには, 要素間のインターフェースをどのように設計するかが重要な課題であるものの, 最終的な性能評価指標である認識率ベースの基準でインターフェースを設計する手法は存在せず, 適当なインターフェースを設け性能評価を繰り返す, といった試行錯誤がなされてきた. しかし近年のアルゴリズムの発達や計算機資源の性能向上により, 複数の要素で構成されたネットワーク全体を一つのニューラルネットワークとしてつなぎ合わせ, その全体を最適化するということが可能になった. このように一連の複数の手続きをつなぎ合わせて一つの手続きとして学習させたモデルのことを End-to-end モデルと呼ぶ. End-to-end モデルは画像のキャプション生成 [18] や機械翻訳 [4] といったタスクで成功を収めている.

End-to-end 音声認識では, 音響特徴量系列を入力, 文字列あるいは音素列を出力とするような

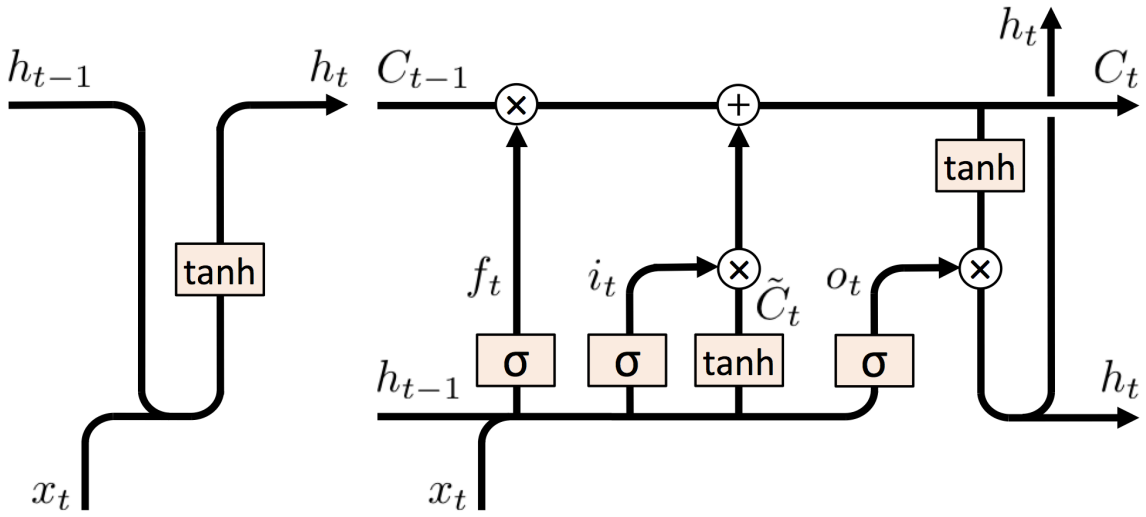


図 2.3: Recurrent Neural Network (左) および Long Short-Term Memory (右) の概略図. RNN は, 活性化関数に tanh を用いた場合の図である. 一方 LSTM については, 各活性化関数はゲートの役割を担うので, 図中の活性化関数をそのまま利用するのが一般的である.

単一のニューラルネットワークを用いて音声認識を行う. このようにして学習を行なうことで認識率ベースの基準によるモデル全体の最適化が可能となる. またネットワークが簡略化されることで, 発音辞書などの専門的な知識を要する要素が明示的には不要となるだけでなく, 複数の要素の出力結果を考慮しなければならない複雑なデコード機構が非常にシンプルなものになる.

End-to-end 音声認識では特徴量系列から文字列等へのマッピングを行うため, ニューラルネットワークの中でも時系列データを扱うことが可能な Recurrent Neural Network (RNN) を拡張させた Long Short-Term Memory (LSTM) が実装に用いられる. このとき, ある文字に対応する音響特徴量は複数フレームにわたるため, この対応関係を学習する必要がある. これを実現させる手法として, 現在提案されている手法は Connectionist Temporal Classification [3] を利用する手法 [1,7,19,20] と Attenstion mechanism を利用する手法 [21–25] の 2 つに大別される.

2.3.2　Long Short-Term Memory

Long Short-Term Memory (LSTM) [26] は, RNN の拡張として提案された, 時系列データに対するニューラルネットワークの構造の一つである. 図 2.3 は RNN および LSTM の模式図である. RNN は各ステップの入力層の情報を中間層に再帰的に入力する構造をとっており, これにより先行するステップに入力された情報を後続ステップに渡すことが可能となっている. しかし, 学習の際に, 入力と逆方向に伝播させた勾配がタイムステップ t に指数関数的に比例して爆発あるいは消失してしまい, 結果として長い系列の学習をうまく行うことが出来ないという問題がある. 勾配が爆発する問題に対しては取りうる値の上限を決めておくクリッピングにより対処可能である. LSTM は勾配消失問題に対する解決策の一つとして提案されたアーキテクチャである.

LSTM には通常の RNN に加え, 入力ゲート, 出力ゲート, 忘却ゲート, メモリセル の 4 つのモジュールが存在する. それぞれのゲートは下式のように隠れ層とあわせてメモリセルの入出力を

制御する役割を持つ.

$$
\begin{aligned}
i_t &= \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) && (2.13)\\
f_t &= \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) && (2.14)\\
o_t &= \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) && (2.15)\\
\tilde{C}_t &= \tanh(W_{Cx}x_t + W_{Ch}h_{t-1} + b_C) && (2.16)\\
C_t &= f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t && (2.17)\\
h_t &= o_t \otimes \tanh(C_t). && (2.18)
\end{aligned}
$$

ここで $\otimes$ は要素ごとの積を求める演算を表す. ニューラルネットワークの勾配消失問題は誤差逆伝播法の過程において活性化関数の微係数が何度も乗算されることによって起こる問題である. RNN では隠れ層が再帰的になってしまっており影響を受けてしまい, 非線形関数の微係数が乗算されてしまうため, 長期依存を学習するのが難しい. 一方 LSTM は, 再帰的な構造をとるノードに関しては恒等関数の活性化関数を用いることで, 勾配消失の影響を受けず, 長期依存を学習することが可能である. これがメモリセル C_t の役割であり, LSTM は時系列タスクでよく用いられる.

一方, 単純な RNN や LSTM ではその構造上, 過去の入力情報しか用いることが不可能である. これを過去から未来へ再帰的に入力するだけでなく, 未来から過去へ再帰的に入力することで得られる隠れ層の 2 つを利用して出力層の計算を行うように拡張した Bi-directional RNN (LSTM) [27] を利用することによってこの問題を解決することが出来る. 具体的には以下のようにして, 前向き隠れ層 $\overrightarrow{h}$ および後ろ向き隠れ層 $\overleftarrow{h}$ から各フレーム t における出力ラベルの確率分布 y_t を得る.

$$
\begin{aligned}
\overrightarrow{h}_t &= \mathrm{LSTM}(x_t, \overrightarrow{h}_{t-1}) && (2.19)\\
\overleftarrow{h}_t &= \mathrm{LSTM}(x_t, \overleftarrow{h}_{t+1}) && (2.20)\\
y_t &= \mathrm{softmax}(W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y). && (2.21)
\end{aligned}
$$

2.3.3　Connectionist Temporal Classification の利用による音声認識

CTC は, 系列のマッピングタスクにおいて HMM を用いずに入力と出力の系列長の違いを吸収し, 両者のアライメント (対応関係) を仮定することなく RNN あるいは LSTM を学習させることが出来る目的関数である. 音声認識においては, CTC は音響特徴量系列 $\boldsymbol{x}$ とそれに対応する文字列あるいは音素列系列 $\boldsymbol{y}$ のアライメントを自動で学習する機構を担う. これは以下のようにして実現される.

いま, $\boldsymbol{y}$ が K 種類のラベルで構成されているとする. CTC は $\boldsymbol{x}$ と $\boldsymbol{y}$ とのマッピングを行う代わりに, $\boldsymbol{x}$ と同じ系列長の中間表現 $\boldsymbol{\pi}$ とのマッピングを行う. $\boldsymbol{\pi}$ は先程の K 種類とブランクラベルの $K+1$ 種類のラベルで構成されており, $\boldsymbol{y}$ に対して同一ラベルの繰り返しとブランクラベルの挿入が許されている. つまり, RNN や LSTM の出力層で中間表現系列 $\boldsymbol{\pi}$ を出力しておき,

- 連続するラベルは 1 つのラベルに短縮する
- ブランクラベルは取り除く

という 2 つの整形ルールを適用させることで, 系列長の違いを吸収したマッピングを実現させている.

このとき, 生起確率 $P(\boldsymbol{y}|\boldsymbol{x})$ は中間表現における複数のパスの生起確率の和であるとする. すなわち, $\boldsymbol{y}$ に対してラベルの繰り返しやブランクラベルの挿入を行って作成できる中間表現の全集合

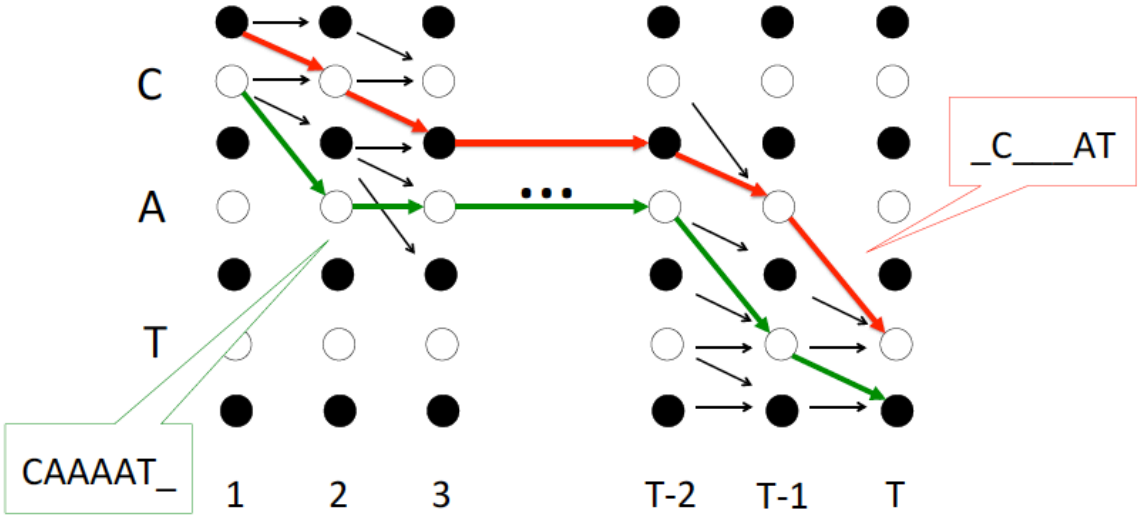


図 2.4: Connectionist Temporal Classification. 黒丸がブランク文字 (ここでは '_' と表記), 白丸が各文字に対応する. 赤色の矢印および緑色の矢印はラベル列 "CAT" に対応するパスの例である. これらのようなパスの生起確率の総和をとることで, ラベル列の生起確率を求めることが可能になる.

を $\boldsymbol{\Phi}(\boldsymbol{y})$ とすると, 生起確率は以下で表される. 図 2.4 はその模式図である.

$$P(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\pi \in \boldsymbol{\Phi}(\boldsymbol{y})} P(\boldsymbol{\pi}|\boldsymbol{x}). \tag{2.22}$$

$P(\boldsymbol{\pi}|\boldsymbol{x})$ は, 各時刻の出力の独立性を仮定することで, それぞれの時刻での出力の生起確率の積で近似される.

$$P(\boldsymbol{\pi}|\boldsymbol{x}) \approx \prod_{t=1}^{T} P(\pi_t|\boldsymbol{x}) = \prod_{t=1}^{T} q_t(\pi_t). \tag{2.23}$$

ここで $q_t(\pi_t)$ は, 時刻 t での RNN の出力 π_t に対する softmax 関数の出力結果である. CTC は損失関数として, 正解ラベル系列 $\boldsymbol{y}$ に対する負の対数尤度で定義され, 以下のように表される.

$$\mathcal{L}_{\mathrm{CTC}}(\boldsymbol{x}, \boldsymbol{y}) \equiv -\ln P(\boldsymbol{y}|\boldsymbol{x}). \tag{2.24}$$

この確率分布を求める際には HMM 同様に forward-backward アルゴリズムを用いて計算量を削減することが可能である.

デコードの際には, 各時刻について最も高い生起確率を与えるラベルを生起することで出力系列を簡単に求めることが可能である. この時ビーム探索を行うことでより正確な出力系列を得ることが出来る.

2.3.4　Attenstion mechanism の利用による音声認識

Attenstion mechanism は, Encoder-Decoder モデル [28] と呼ばれる 2 つの RNN を結合させたモデル構造によって実現される. 一方の RNN/LSTM (Encoder) では, 特徴量系列 $\boldsymbol{x}$ を入力し中

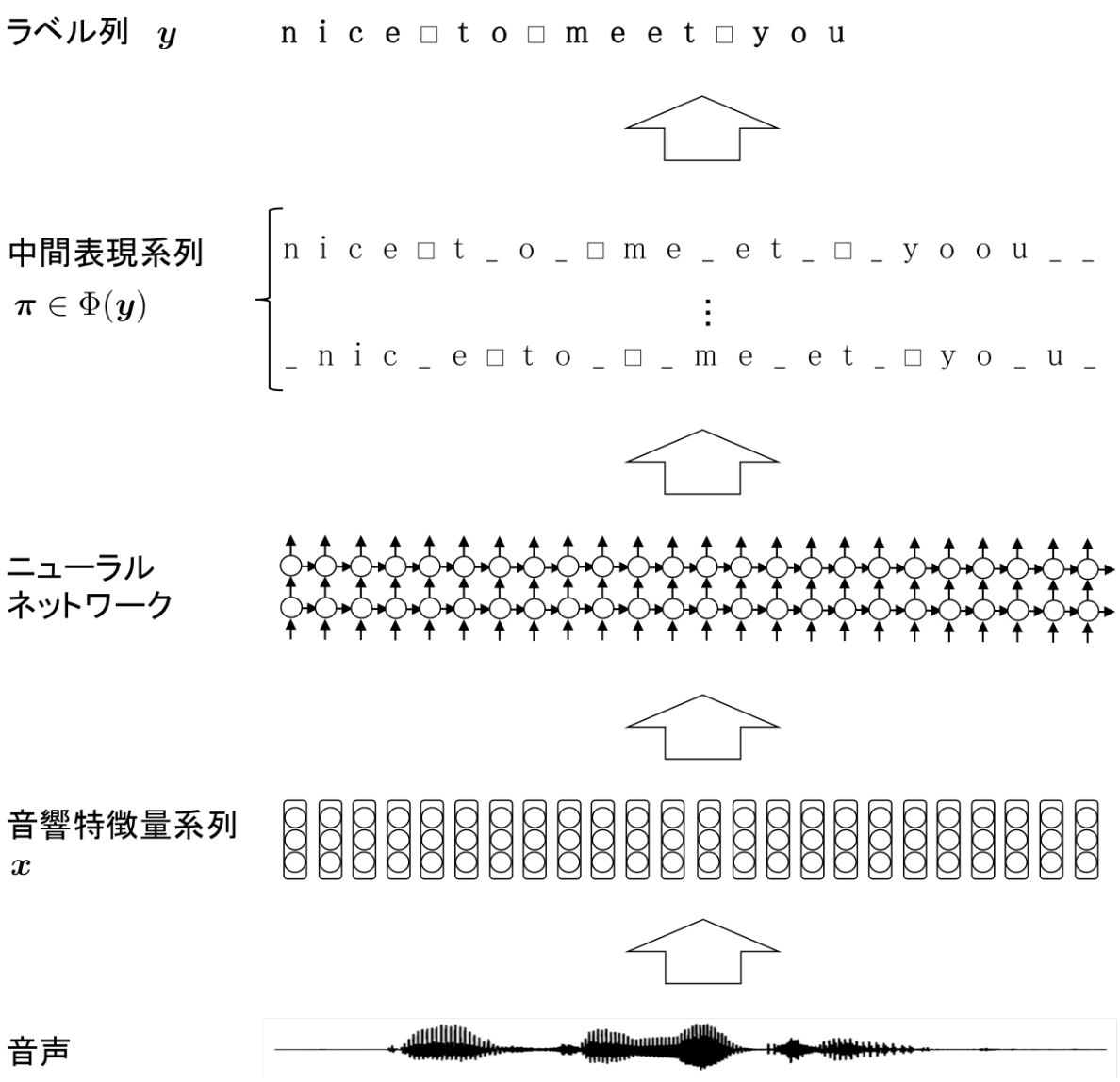


図 2.5: CTC の利用による End-to-end 音声認識の枠組み

間表現系列 $\boldsymbol{h}$ を出力させる．そしてもう一方の RNN/LSTM (Decoder) で, 中間表現系列 $\boldsymbol{h}$ から所望のラベル系列 $\boldsymbol{y}$ を出力させる．この Encoder と Decoder を同時学習することによって, 音響特徴量系列 $\boldsymbol{x}$ から文字列 $\boldsymbol{y}$ を直接出力させる End-to-end 音声認識を実現させている．このとき, 以下で定式化されるような Attention mechanism [4] (以下 Attention) と呼ばれる注意機構を用いて, 毎度の出力の際に入力系列中のどのあたりを参照すべきかを考慮しながら出力を行うのが

このモデルの特徴である.

$$
\begin{aligned}
\boldsymbol{h} &= \mathrm{Encoder}(\boldsymbol{x}) && (2.25)\\
\boldsymbol{f}_u &= F * \alpha_{u-1} && (2.26)\\
e_{u,l} &= \boldsymbol{w}^T \tanh(W\boldsymbol{s}_{u-1} + V\boldsymbol{h}_l + U\boldsymbol{f}_{u,l} + b) && (2.27)\\
\alpha_{u,l} &= \frac{\exp(e_{u,l})}{\sum_l \exp(e_{u,l})} && (2.28)\\
\boldsymbol{c}_u &= \sum_l \alpha_{u,l}\boldsymbol{h}_l && (2.29)\\
\boldsymbol{y}_u &\sim \mathrm{Generate}(\boldsymbol{c}_u, \boldsymbol{s}_{u-1}) && (2.30)\\
\boldsymbol{s}_u &= \mathrm{Recurrency}(\boldsymbol{s}_{u-1}, \boldsymbol{c}_u, \boldsymbol{y}_u). && (2.31)
\end{aligned}
$$

すなわち Decoder は, Encoder の出力の重み付け和 $\boldsymbol{C}_u$ および前の入力に対する Decoder の隠れ層 $\boldsymbol{s}_{u-1}$ を入力として受け付け, ラベルの確率分布 $\boldsymbol{y}_u$ および新しい Decoder の隠れ層 $\boldsymbol{s}$ を出力する, という処理を再帰的に行なう. 重み付け係数 $\alpha_{u,l}$ は, Encoder の出力および前の入力に対する Decoder の隠れ層 $\boldsymbol{s}_{u-1}$, 前の入力時の重み付け和 $\alpha_{u-1,l}$ によって決められる. 上式中の $\boldsymbol{w}, W, V, F, U, b$ は学習によって決まる Encoder-Decoder モデル中のパラメータである. 終了ラベルが出力されるまでの出力を出力ラベル列とみなし, 正解ラベルによるクロスエントロピー誤差基準によって学習を行なうことで, End-to-end 音声認識が実現される.

2.3.5　End-to-end 音声認識システムの比較

Attention ベースの手法は, CTC ベースの手法とは異なり各ステップごとの出力に独立性を仮定していないため, 言語モデルがない場合には CTC ベースのものよりも精度が高い [23]. また, CTC ベースの手法では時系列的に一方通行的なアライメントしか受け付けないのに対し, Attention ベースの手法では柔軟にアライメントを考慮することが可能である. しかし Attention ベースの手法では, 何も制約を課さない場合, 入力系列が長くなるにつれ捉えるアライメントにずれが生じる可能性が高まる. これに対処すべく, 入力系列すべてを対象にするのでなく重み付け対象区間を狭める方法 [23] が提案されているが, この対処法では学習データに依存して人手で窓区間の長さを決める必要がありハイパーパラメータが増加してしまうという欠点が挙げられる.

一方, Attention mechanism は雑音等によって容易に崩れてしまうため, Attention ベースの手法は CTC ベースの手法に比べ雑音に弱い [6]. また前述にも述べたように Attention ベースの手法に関して, 入力系列が長くなるにつれアライメントに自由度が増すことにより学習が難化するため, 特にスクラッチでの学習は CTC ベースの手法の方が早く収束する [6].

以上のように, CTC ベースの手法と Attention ベースの手法は一長一短な点があるが, 両者を組み合わせて学習を行うモデルも近年提案されている [6, 29]. これらのモデルは Attention ベースの手法と比べて雑音に頑健なだけでなく, 収束も早く認識精度も高いため注目を集めている.

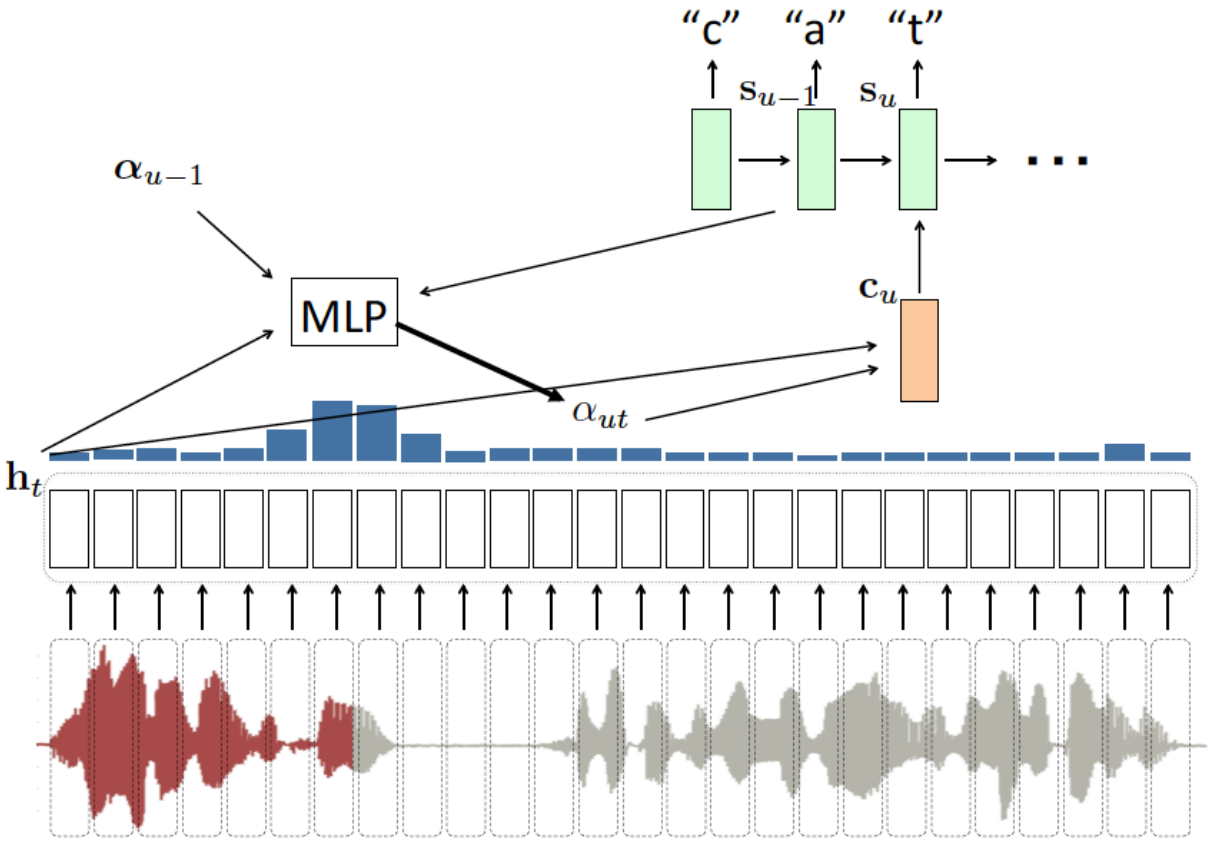


図 2.6: Attenstion mechanism の利用による End-to-end 音声認識の枠組み

第3章

敵対的学習

3.1　はじめに

DNN は画像認識や音声認識等の分野で驚異的な精度を達成しているが, 非直感的な認識を行うことがあり, このような敵対的サンプルを意図的に生成することが可能である. このようにして生成された敵対的サンプルをも対象に学習を行なうのが敵対的学習であり, これにより汎化性能を向上させることが可能である. 本章では敵対的サンプルの意味や敵対的学習の目的についてより詳細な説明を行なったのち, 本研究で扱う勾配に基づく敵対的学習の手法の説明を行なう.

3.2　敵対的サンプルおよびそれに基づく敵対的学習の概要

DNN は音声認識や画像認識といった分野をはじめとした様々な分野で高い精度を達成しているが, 勾配を逆方向に伝播させる手法によってモデルパラメータの更新を行なっているために反直感的な性質を持つことがあるという報告がある [10]. 特に, state-of-the-art を達成している画像認識タスクにおいて, 特定の手法を用いて求めた小さな摂動を加えて新しい画像を生成すると, 人間には元画像と同じように見えるのに高い確信度で誤分類することがある. このようにして生成されたサンプルは敵対的サンプル (adversarial examples) と呼ばれている. このような敵対的サンプルは元のサンプルの近傍に存在しているため, 本来であれば元のサンプルと同様の出力が行われることが期待されるが, 高い確信度で誤分類を起こしてしまうことから, 機械学習の学習手法や実用化に関して不十分な点があることが示唆されている.

図 3.1 は [10] で挙げられている敵対的サンプルの例である. これらの例は以下で示すような Box-constraind L-BFGS という手法で生成されている. なお, x は入力, r は摂動, l は対象となるラベル, $c > 0$ は定数, m は入力及び摂動の次元数である. また Loss は入力と正解ラベルから決まる誤差基準である.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c|r| + \text{Loss}(x + r, l) \\ \text{subject to} \quad & x + r \in [0, 1]^m. \end{aligned}$$

この論文中では敵対的サンプルの持つ性質として,

- あるモデルの敵対的サンプルは, 別のアーキテクチャのモデルでもよく誤分類される
- あるデータで学習したモデルの敵対的サンプルは, 別のデータで学習したモデルでもよく誤分類される

といったものも挙げられており, 敵対的サンプルが生まれる原因が特定のモデルアーキテクチャや特定のデータセットによる過学習ではないことが示されている.

Goodfellow らは敵対的サンプルが生まれる原因について, 以下のような説明を行なっている [8]. 入力 $\boldsymbol{x}$ に摂動 $\boldsymbol{\eta}$ を加えたサンプル $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\eta}$ について, ニューラルネットワーク中のある重み $\boldsymbol{w}$ との内積を考えると,

$$\boldsymbol{w}^T \tilde{\boldsymbol{x}} = \boldsymbol{w}^T (\boldsymbol{x} + \boldsymbol{\eta}) = \boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{w}^T \boldsymbol{\eta} \tag{3.1}$$

となる. この時, 第 2 項が大きくなればなるほど摂動加算前の入力に対する出力が変形されてしまう. 具体的には $\boldsymbol{\eta} = \text{sign}(\boldsymbol{w})$ とすることで最大の値を取るようになる. $\boldsymbol{\eta}$ の各要素の値が微小な値 ϵ 以下であるとしても, 入力 $\boldsymbol{x}$ が高次元であればあるほど第 2 項は大きくなるため, 例えば画像認識タスクにおいて, 入力画像からはほんの僅かな変化しか観測されないがニューラルネットワークの出力は大きく異なる, という現象が実現可能である.

図 3.1: 敵対的サンプルの例 [10]．正しい分類が可能な 6 枚の画像（左）に対し, モデルが誤識別するように意図的に生成された摂動（中央）を加えることによって敵対的サンプル（右）が生成されている．生成された新しい画像は全て “ダチョウ” と認識される．

Goodfellow らはさらに, ニューラルネットワークの学習において敵対的サンプルも学習データに含めて学習を行う敵対的学習 (Adversarial training) を提案している [8]．敵対的学習ではサンプルに摂動を加えた場合でも同じ出力をするように学習を行うためモデルの頑健性の向上が期待されるが, それだけでなく, 元のテストデータにおける汎化性能も向上することから, 新たな正則化手法としても注目されている．

3.3　Adversarial training

Goodfellow らは, 線型性に着目して Fast Gradient Sign Method (FGSM) と呼ばれる敵対的サンプルを高速に生成する手法に基づいた敵対的学習を提案している [8]．以降ではこの手法を Adversarial training (AT) と呼ぶことにする．

x を入力, y を正解ラベル, θ をモデルのパラメータとする．損失関数は以下のように定義され, この値が小さくなるように θ の学習が行われる．

$$E(x, y, \theta) = -\log P(y|x; \theta). \tag{3.2}$$

AT では, 正解ラベルの尤度が最も小さくなるような摂動 r_{at} を考慮し, これを入力 x に加えたものを敵対的サンプルとする．

$$r_{at} = \underset{r, |r| \leq \epsilon}{\operatorname{argmin}} \log P(y|x + r; \hat{\theta}). \tag{3.3}$$

ここで $\hat{\theta}$ は学習時のモデルパラメータである．

しかし式 (3.3) は制約付き最適化を伴い, 敵対的サンプルの生成やモデルの学習を繰り返す必要があるために解析的に解くのは非効率である．そこで以下のような線形近似が提案された．

$$r_{at} \approx \epsilon \operatorname{sign}(\nabla_x E(x, y, \hat{\theta})). \tag{3.4}$$

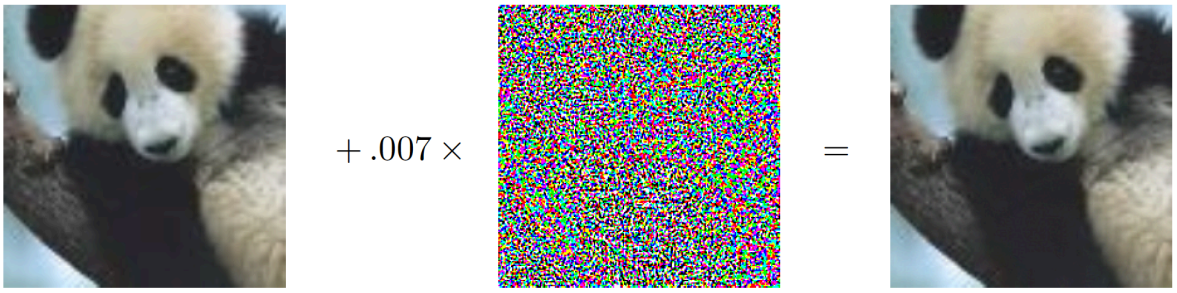


図 3.2: Adversarial training によって生成される敵対的サンプルの例 [8]. パンダの画像と, Adversarial training によって生成される摂動を加えて生成された新しい画像とでは, 人間が目で見た場合には違いが感じられないが, 識別器は高い確信度で “テナガザル” と誤認識するようになる.

式 (3.4) 中の勾配は, モデルがニューラルネットワークの場合は誤差逆伝播により一度だけ勾配計算を行えばよいので簡単に求めることが可能である. このように損失を最大化させるような方向に入力 x を変化することで, 確実ではないが高速に敵対的サンプルを生成することが可能となった. 図 3.2 はこの手法によって生成された敵対的サンプルの例であり, 以上のような近似式によって高速化を行なった場合でも敵対的サンプルが高確率で生成できるようになる.

AT ではモデルの損失関数を以下のように置き換えて行うことで, 敵対的サンプルに対する頑健性を同時に学習する.

$$\mathcal{L} = E(x, y, \hat{\theta}) + \alpha E(x + r_{at}, y, \hat{\theta}). \tag{3.5}$$

ここで α は本来の損失関数と敵対的サンプルに対する尤度ベースの損失関数の重み付けを行うためのハイパーパラメータであるが, $\alpha = 1$ が推奨されている.

3.4　Virtual adversarial training

前述の AT が正解ラベルの尤度を下げるような摂動を考慮していたのに対し, 宮戸らが提案した Virtual adversarial training (VAT) [9] では該当するラベルに対する確率だけでなく, ラベル全体に対する確率分布が最も変わるような摂動を考える. より具体的には, 摂動を加えない場合の出力分布と加えた場合の出力分布との KL ダイバージェンスが小さくなるような方向への学習を行うことを目的として, 以下で表される項を目的関数に加える.

$$\mathcal{L}_{\text{VAT}} \equiv \Delta_{\text{KL}}(r_{vat}, x) \tag{3.6}$$

ただし

$$\Delta_{\text{KL}}(r, x) = \text{KL}[P(y|x; \hat{\theta}) | P(y|x + r; \hat{\theta})] \tag{3.7}$$

$$r_{vat} = \underset{r, |r| \leq \epsilon}{\text{argmax}}\, \Delta_{\text{KL}}(r, x). \tag{3.8}$$

ノイズの混入に対してモデルの出力分布が変わらないように学習されるため, VAT の利用によって予測分布がなめらかになることが期待される. また VAT によって算出される正則化項は正解ラベルを必要としないため, 半教師あり学習に応用することが可能である.

AT 同様, r_{vat} を解析的に求めるのは難しく非効率であるため, 以下のような効率的な近似手法を用いて高速化を行なっている.

$r = \mathbf{0}$ まわりのテイラー展開を用いた 2 次近似によって

$$\Delta_{\mathrm{KL}}(r, x, \theta) \simeq \Delta_{\mathrm{KL}}(\mathbf{0}, x, \theta) + r^T \nabla_r \Delta_{\mathrm{KL}}(r, x, \theta)|_{r=\mathbf{0}} + \frac{1}{2} r^T H(x, \theta) r \tag{3.9}$$

と表現できる. ここで H はヘッセ行列であり, $H(x, \theta) = \nabla\nabla_r \Delta_{KL}(r, x, \theta)|_{r=\mathbf{0}}$ である. $r = \mathbf{0}$ のとき, モデルの出力分布は等しくなるため KL ダイバージェンスは 0 になる. また KL ダイバージェンスは距離尺度であり, $r = \mathbf{0}$ で最小値をとるため, $r = \mathbf{0}$ における勾配も 0 となる. 従って式 (3.9) は

$$\Delta_{\mathrm{KL}}(r, x, \theta) \simeq \frac{1}{2} r^T H(x, \theta) r \tag{3.10}$$

となる.

一般に $|x| = 1$ のとき, $x^T A x$ の最大値は行列 A の最大固有値となるので, H の最大固有値に対応する固有ベクトルを u と表せば,

$$\begin{aligned} r_{vat} &= \underset{r}{\mathrm{argmax}} \{\Delta_{\mathrm{KL}}(r, x, \theta); ||r||^2 < \epsilon\} && (3.11) \\ &\simeq \epsilon \overline{u(x, \theta)} && (3.12) \end{aligned}$$

と近似できる. ここで $\overline{x}$ は x を単位ベクトル化したものである.

以上より H および $\boldsymbol{u}$ を算出することで近似的に $\boldsymbol{r}_{vat}$ を求めることが出来るが, べき乗法および有限差分法を用いることでさらに高速化のための近似を行うことを考える.

べき乗法は行列の最大固有値および対応する固有ベクトルを求める方法である. あるベクトル $\boldsymbol{d}$ に以下の演算を繰り返し適用することを考える.

$$d \leftarrow \overline{Hd}. \tag{3.13}$$

これにより d は行列 H の最大固有値に対応する固有ベクトルに漸近する. 以上によって近似的に u を求めることが可能である.

一方で Hd そのものの導出には以下のようにして有限差分法を用いることで計算コストを抑えることが可能である.

$$Hd \simeq \frac{\nabla_r \Delta_{KL}(r, x, \theta)|_{r=\xi d} - \nabla_r \Delta_{\mathrm{KL}}(r, x, \theta)|_{r=\mathbf{0}}}{\xi d} d$$

ここで, 分子第二項は $r = \mathbf{0}$ の時の KL ダイバージェンスの微分値である. 前述の通り, $r = \mathbf{0}$ の時の KL ダイバージェンスおよびその微分値は 0 であるから,

$$Hd \simeq \frac{\nabla_r \Delta_{\mathrm{KL}}(r, x, \theta)|_{r=\xi d}}{\xi} \tag{3.14}$$

となる. この分子の項はネットワークの順伝播, 逆伝播を行うことで求めることができる.

以上より, べき乗法によって

$$d \leftarrow \overline{\nabla_r \Delta_{KL}(r, x)|_{r=\xi d}} \tag{3.15}$$

を繰り返し行なって得られた d を用いて,

$$r_{adv} = \epsilon d \tag{3.16}$$

として近似値を得ることが可能である. また [9] によると, べき乗法の適用回数は 1 回で十分とされており, さらに計算量を抑えることが可能である.

VAT の学習についても, AT 同様に以下のように敵対的サンプルに対する頑健性を同時に学習する.

$$\mathcal{L} = E(x, y, \hat{\theta}) + \alpha \Delta_{KL}(r_{adv}, x, \theta). \tag{3.17}$$

このときハイパーパラメータは, 損失関数の重み α・r_{adv} のスケール ϵ・r_{adv} の近似を行う際に必要となる微小値 ξ の 3 つだが, $\lambda_1(x)$ を $H(x)$ の最大固有値とすると,

$$\begin{aligned} \max r \Delta_{KL}(r, x, \theta) &\simeq \max r \frac{1}{2} r^T H(x, \theta) r & (3.18) \\ &= \frac{1}{2} \epsilon^2 \lambda_1(x) & (3.19) \end{aligned}$$

が成立することから, α と ϵ の調整はどちらか一方のみ行えばよい. 従って $\alpha = 1$ と固定してしまっても差し支えない. また ξ に関しても 10^{-6} 程度の微小値を選んでおけばよい. したがって ϵ についてのみ開発データを用いて決定すればよく, 少ないハイパーパラメータで VAT を導入することが可能である. また VAT は正解ラベルが付いてないデータも学習に利用することが出来るため, 半教師あり学習として導入することも可能である.

第4章

End-to-end音声認識における敵対的学習の適用

4.1　はじめに

End-to-end 音声認識は近年盛んに研究されており, その精度は従来の音声認識と同程度あるいはそれ以上である [7,29,30]. Amodei らの研究 [7] では, 数千時間もの大量の音声を学習に用いることで, 静音環境下での英語および中国語の音声認識において人間と同程度の認識率を達成した. 堀らの研究 [29] では, CTC 型と Attention 型を融合させた新しいアーキテクチャを提案し, 日本語の音声認識において言語モデルを用いることなく従来の音声認識を超える精度を達成した.

しかしながら, 構成要素を別々のコーパスで学習させていた従来の音声認識システムとは異なり, End-to-end 音声認識では一つのコーパスを元に学習が行われるため, コーパスの性質を受けやすい傾向にある. 特に耐雑音性は従来の音声認識でも人間に遠く及ばず [5] 改善の余地があり, 例えば clean な音声で構成されているコーパスを使って学習させた場合には当然のように耐雑音性は著しく低くなる. 本研究では, 耐雑音性を汎化性の一つとして捉え, 敵対的学習の枠組みを導入することによって End-to-end 音声認識の汎化性能を向上を行なった. 特に, 敵対的サンプルを生成する枠組みは音声に雑音を付与する処理とみなすことが出来る. この手法によって付与される雑音は種類を全く仮定する必要がなく, 学習の各ステップで毎度自動で算出されるため, 耐雑音性の向上に繋がることが期待される. またこの枠組みでは学習データの明示的な増量が不要であるため, ネットワークの最適化の難化といった問題を避けることが可能である.

本章では, 敵対的学習を適用させた End-to-end 音声認識の枠組みについて説明を行なったのち, 音素認識実験や音声認識実験によって手法の評価を行なう. なお End-to-end 音声認識を行なうにあたり, 雑音環境下での音声認識に関しては, 雑音によって Attention mechanism の働きが悪化し精度が大きく落ちるとの報告がある [6] ため, 本研究では CTC を用いた End-to-end 音声認識のみを扱った. また CTC の出力だけでなく言語モデルや語彙情報等を利用することで精度を向上させることが出来るため, 音声認識システムを構成する際にはこれらを使用するのが一般的だが, 今回は使用しなかった.

4.2　提案手法

本研究では, CTC の枠組みに敵対的学習として AT および VAT を適用させることによって End-to-end 音声認識の正則化を行なった. AT および VAT は元々画像認識の分野で考案された手法だが, これを音声認識という系列タスクに適用させた. CTC への適用は毎時刻ノイズを生成することによって成立し, 以下のように目的関数を変更することで可能となる.

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{\mathrm{CTC}}(\boldsymbol{x}, \boldsymbol{y}) + \alpha \mathcal{L}_{\mathrm{AT}}(\boldsymbol{x}, \boldsymbol{y}) \\ &= -\ln P(\boldsymbol{y}|\boldsymbol{x}) - \alpha \ln P(\boldsymbol{y}|\boldsymbol{x} + \epsilon \mathrm{sign}(\nabla_{\boldsymbol{x}} \ln P(\boldsymbol{y}|\boldsymbol{x}))). \end{aligned} \tag{4.1}$$

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{\mathrm{CTC}}(\boldsymbol{x}, \boldsymbol{y}) + \alpha \mathcal{L}_{\mathrm{VAT}}(\boldsymbol{x}) \\ &= -\ln P(\boldsymbol{y}|\boldsymbol{x}) - \alpha \sum_t \Delta_{\mathrm{KL}}(\epsilon \overline{\nabla_r \Delta_{KL}(r, x_t)|_{r=\xi d}}, x_t). \end{aligned} \tag{4.2}$$

ここで α は正則化項の重み付けのための値, ϵ はノイズの規模を表す値, d はランダム値を要素とする単位ベクトルである. α と ϵ はハイパーパラメータである.

敵対的学習は前述のようにモデルの性能を悪化させるノイズを考慮して学習を行う手法であるため, 耐雑音性のような性能を向上させることができると期待される.

表 4.1: TIMIT 音声コーパスの内訳

項目	話者数	発話数
学習データ	462	3,696
開発データ	50	400
評価データ	24	192

表 4.2: 音響分析条件

項目	設定値
サンプリング	16 bit / 16 kHz
窓	25 ms 幅 / 10 ms シフト
音響特徴	対数メルフィルタバンク (40 次元) + Δ・ΔΔ 特徴量 (計 120 次元)

4.3　小規模コーパスを用いた音素認識

4.3.1　実験設定

実験に用いる音声コーパスには, TIMIT データベース [31] を用いた. TIMIT データベースには音素がバランスよく並んだ英語文の読み上げ音声が収録されており, 音響モデルの構築や音声認識あるいは音素認識タスクでよく用いられる. 630 人の話者が 10 文ずつ読み上げた計 6,300 文で構成されているが, 実験で用いられる際には一部の方言話者による発話を除き表 4.1 のような設定で用いられることが一般的である. 今回の実験でもこの設定に従った.

音響分析には Kaldi Speech Recognition Toolkit [32] を利用し, 表 4.2 のような条件で音響特徴量を抽出した. 正解ラベルには 39 個の音素を用いた. また TIMIT コーパスの音声は雑音が混入しない clean な音声で構成されているため, 雑音環境化での性能を評価することを目的として, 評価データに対し雑音を付与した noisy な評価データを用意した. 雑音は 100 Nonspeech Sounds[1] 中の以下 4 種類を使用し, SNR が 20, 15, 10, 5, 0, -5dB となるようにそれぞれ雑音を付与した.

- Crowd noise
- Machine noise
- Alarm and siren
- Traffic and car noise

End-to-end 音声認識を行なうニューラルネットワークのアーキテクチャには Bidirectional-LSTM を使用した. 実装には TensorFlow[2]を利用した (これは以降の実験でも同様である). パラメータや学習条件は表 4.3 の通りである. AT や VAT を適用する上で, 式 (4.2) および式 (4.3) 中のノイズの規模 ϵ を決めるにあたり, 開発データを用いてグリッドサーチを行ない, 最も認識精度が高かったものを用いた. AT については [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 1.0, 3.0] の範囲でグリッドサーチを行い, 0.3 とした時に最も高い精度が得られた. 一方 VAT については [3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 10.0, 30.0] の範囲でグリッドサーチを行い, 5.0 とした時に最も高い精度が得られた. 式 (4.2), 式 (4.3) 中の AT および VAT の正則化項の重み α は 1.0 とした.

さらに, AT および VAT によって生成されるノイズと比較するために, ガウスノイズを毎時刻生

[1] http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/
[2] https://www.tensorflow.org/

表 4.3: TIMIT 音素認識実験における Bidirectional-LSTM のパラメータおよび学習条件

パラメータ	値
層数	1
次元数	256
重みの初期値	-0.1 〜 0.1 の一様乱数
最適化手法	Adam [33]
初期学習率	0.0005
バッチサイズ	32
CTC デコード時のビーム幅	20

表 4.4: clean な TIMIT 開発データ・評価データに対する PER

手法	PER(開発データ)	PER(評価データ)
CTC (ベースライン)	24.40	26.72
CTC + random noise	24.74	26.57
CTC + AT	**22.73**	**23.86**
CTC + VAT	22.77	24.57

成し入力特徴量に付与した上で学習を行なうモデルを用意した. AT および VAT で付与されるノイズの大きさが同程度になるように, 加えるガウスノイズの標準偏差は 0.3 とした.

4.3.2　実験結果

静音環境下における認識精度は表 4.4 のようになった. AT および VAT を適用させたモデルは, 通常の CTC によって得られる対数尤度のみで学習させたベースラインよりもエラー率が低く, AT/VAT の両手法は開発データに対して 6.8/6.6%, 評価データに対して 10.7/8.0%の相対改善率を達成した. AT と VAT とで比較した場合は, AT の方が VAT を上回った. AT は VAT と異なり, 正解ラベルを用いた正則化を行なっているため, 正解ラベルの出力確率がよりシャープになるように学習される傾向にあると考えられる. 静音環境下でモデルを学習させた場合には, ノイズがなく明瞭な分, 各音素がより明確に区別されるため, AT の性質によってより高い対数尤度が得られるようになり, VAT を上回る結果が得られたのだと考えられる.

図 4.2 は, 雑音を付与して得られた評価データに対する認識結果である. まずはじめに, 静音環境下での結果 (表 4.4) と比べると特に SNR が小さい場合に認識率が大幅に悪化しているが, これは正解ラベルに無音区間を表すラベル 'sil' がある一方で, 雑音が付与されたために無音区間の判別が難しくなってしまったからだと考えられる. 次にそれぞれの雑音について, machine noise および traffic and car noise に関しては, AT および VAT がベースラインを大きく上回る結果を達成したが, crowd noise および alarm and siren に関してはモデル間であまり差がつかない結果となった. crowd noise は人間の音声が入り混じったノイズであり入力音声に近い性質を持つため, AT および VAT では改善させることは難しいと考えられる. alarm and siren に関しては若干の精度改善が見られたものの, machine noise および traffic and car noise よりも改善幅が小さい. AT および VAT によって総じて耐雑音性を向上させることができたが, ノイズの種類によっては効果が出にくいものもあるという結果が得られた.

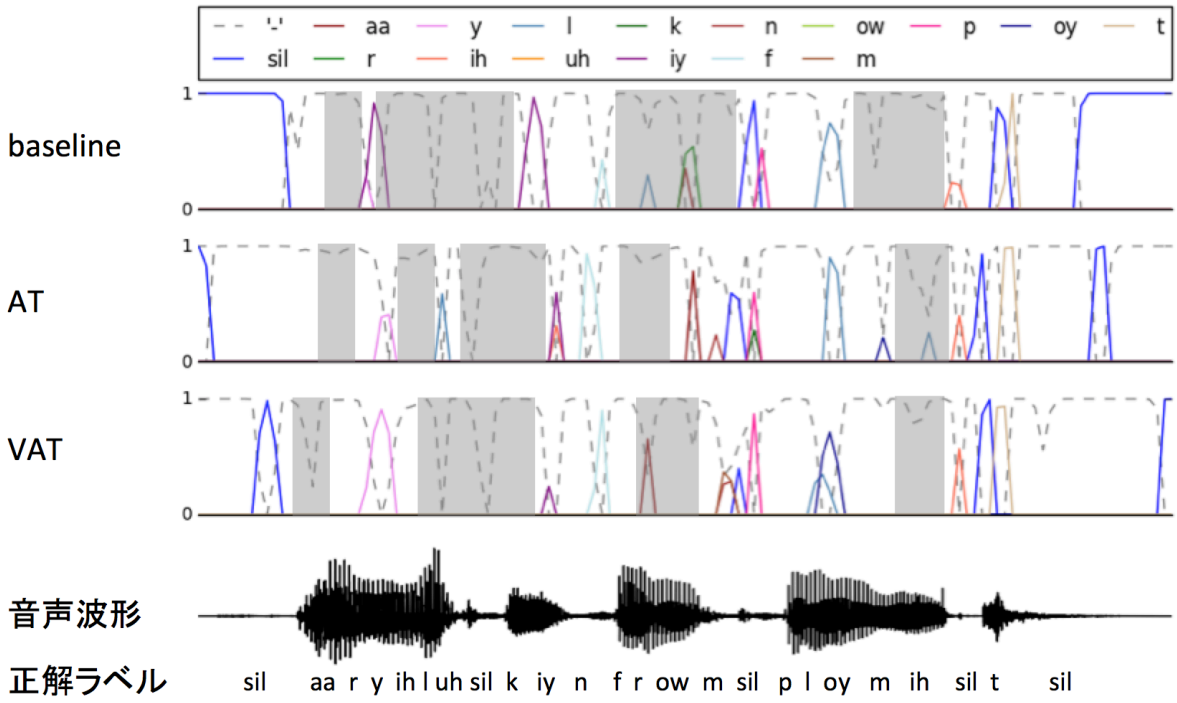
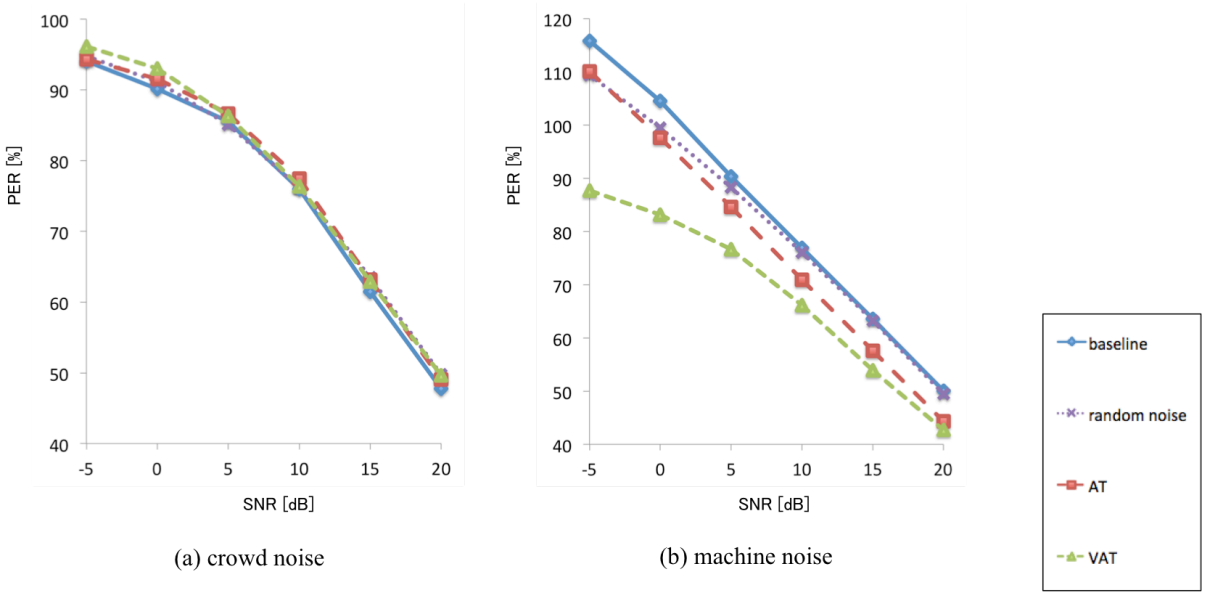
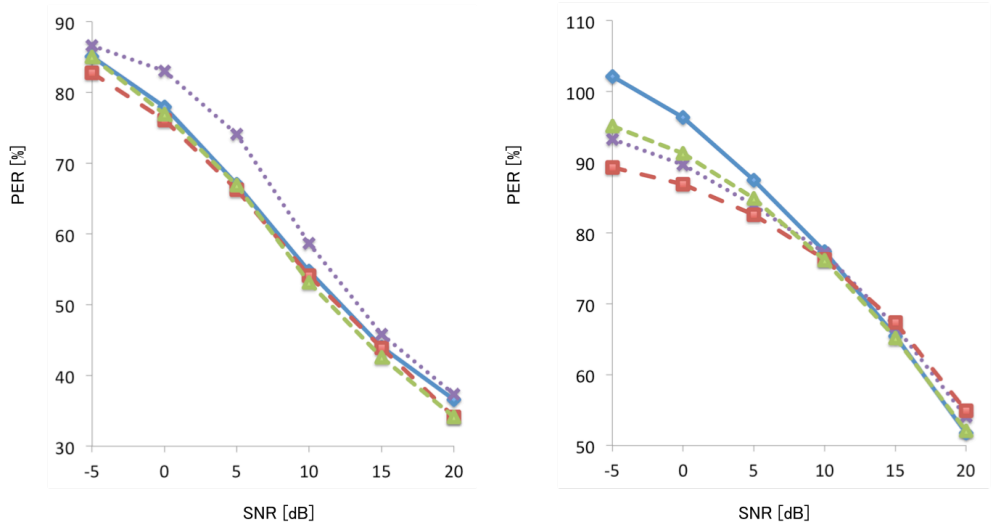


図 4.1: TIMIT 音素認識におけるフレームごとの音素の確率分布の例．各手法に対して, CTC の出力ので最も尤度の高い音素が色ごとに図示されている．なお '-' は CTC のブランク文字, 'sil' は無音区間を表すラベルである．また誤認識が起きている箇所は灰色の網掛けで示されている．なお対象としている発話の書き起こし文は “Are you looking for employment?” である．



(a) crowd noise　(b) machine noise



(c) alarm and siren　(d) traffic and car noise

図 4.2: noisy な TIMIT 評価データに対する各モデルの認識結果

表 4.5: WSJ0 および Aurora4 音声コーパスの内訳

項目	WSJ0		Aurora4	
	データセットのラベル	発話数	データセットのラベル	発話数
学習データ	-	7,138	-	7,138
開発データ	dev93	503	dev330	4,620
開発データ	eval92	333	test166	2,324

表 4.4 および図 4.2 では, 比較のためにガウシアンノイズを付与して学習させたモデル (CTC + random noise) の結果も共に示されている. このモデルは条件によってはベースラインよりも低い PER を達成している場合もあるが, 静音環境下・雑音環境下の両方において AT および VAT ほどの精度改善には至っていない. これは単にノイズを付与すればよいという訳ではないということを示しており, 同時にこれにより, AT および VAT によって生成されるノイズには大きな耐雑音性向上効果があるということが示された.

図 4.3 は, 評価データから (a) ランダムに選ばれた発話のメルフィルタバンク特徴量, (b)/(c) その発話に対して AT/VAT によってノイズを付与されたメルフィルタバンク特徴量, (d)/(e) crowd noise / machine noise を SNR が 10 dB になるように付与した音声のメルフィルタバンク特徴量である. この図からわかる通り, AT や VAT によってノイズを加えられた (b) および (c) は, 何も加えられていない (a) と比べてほとんど差が見られない. これは 3.2 章 で登場した画像の例でも同様であった. というのも, (b) や (c) で付与されたノイズの規模 ϵ は, 開発データを用いた最適化の結果小さな値に設定されているためである. 従って, このノイズの規模を操作し (d) や (e) のように大きなノイズを付与した上で学習を行うことによって, さらなる耐雑音性向上が可能になると期待される. ところがそのような大きな規模の ϵ に設定した上で学習を行なった場合, 認識精度は静音環境下・雑音環境化の両者において悪化した. これは, ノイズの探索範囲が $|r| \leq \epsilon$ とノイズのノルム基準になっており, 音響特徴量空間での距離基準と必ずしも合致しないため, 生成するノイズを大きくすることで学習が難化したためだと考えられる. 例えば同一話者の異なる音素の特徴量間の距離と異なる二話者の同一音素間の特徴量間の距離を比較した場合, 前者の方が小さく場合がほとんどである. 特に後者に関して, 性別が異なる場合には, 同じラベルが割り振られているとはいえ特徴量間の距離は大きなものになる. 従って, ある音声サンプルに対してモデルが誤認識しやすいノイズを探索させる場合には, 以上のような性質に注意した探索空間を考慮すべきだと考えられる. また, AT や VAT のノイズ付与の過程に関して, 入力 x に対してノイズを付与させる場合に今回は単純に加算させて $x + r$ としていたが, 入力の音響特徴量がスペクトルであることを考えるとこの処理は周波数領域における加算になるため, 時間領域では乗算あるいは畳み込みに相当する. 今回の評価で用いた音声データは時間領域における加算によってノイズが付与されているため, 今回のようなノイズ付与の枠組みは適していなかったと考えられる. 一方で, ノイズの種類には加算性ノイズと乗算性ノイズ等さまざまなものがあるため, より多くのノイズに対して高い耐雑音性を実現するためにはこれらを考慮した適用方法に変更する必要があると考えられる.

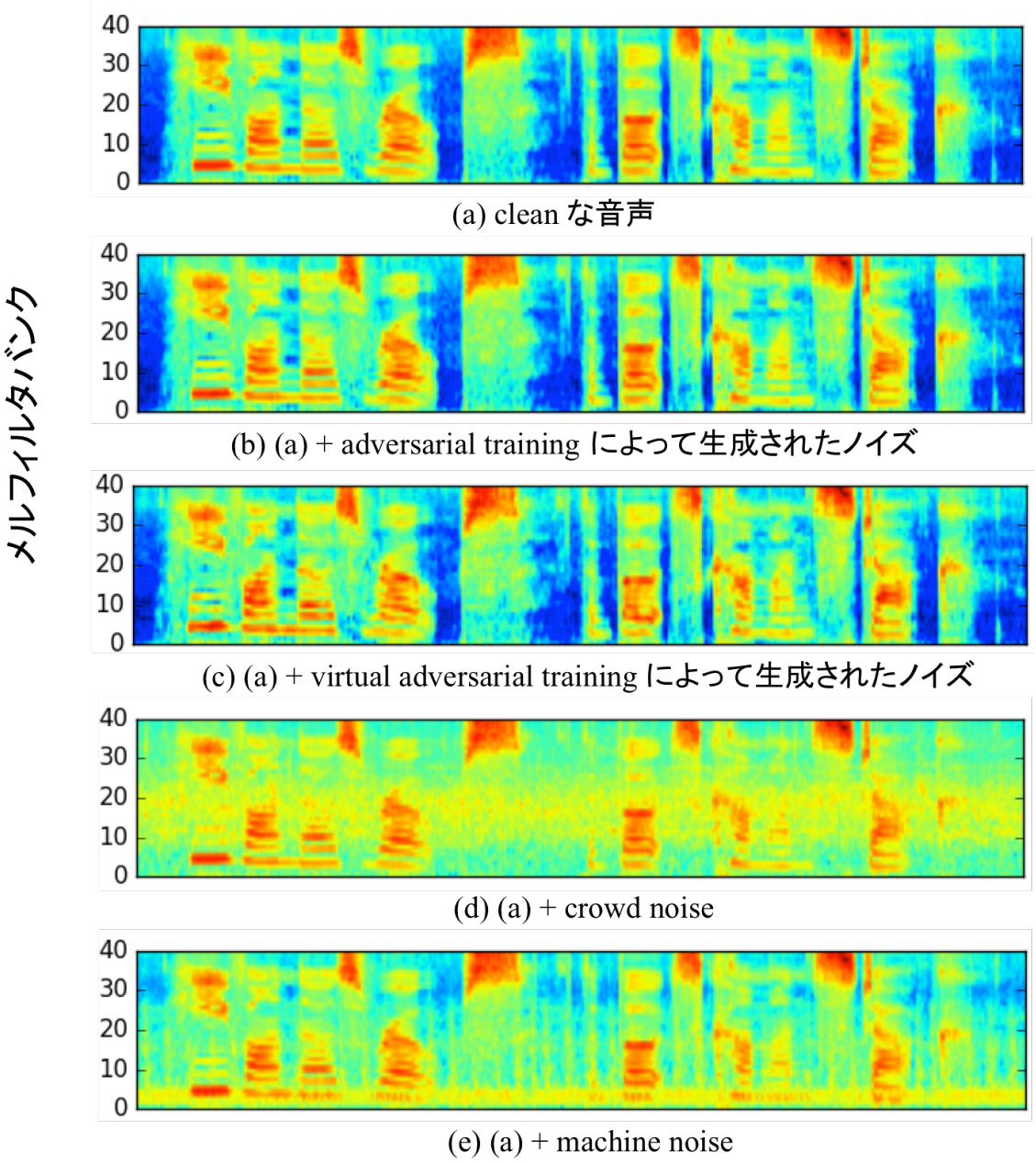


図 4.3: ノイズが付与されたメルフィルタバンク特徴量の例. ここでの “ノイズ” は, 実際の環境音のようなノイズだけでなく, AT や VAT によって生成されたノイズも含む.

表 4.6: WSJ0/Aurora4 音声認識実験における Bidirectional-LSTM のパラメータおよび学習条件

パラメータ	値
層数	4
次元数	256
重みの初期値	-0.1 〜 0.1 の一様乱数
最適化手法	Adam [33]
勾配の最大値の絶対値	10
初期学習率	0.001
バッチサイズ	16
CTC デコード時のビーム幅	20

4.4　中規模コーパスを用いた音声認識

4.4.1　実験設定

今回の実験では, clean な音声コーパスとして Wall Street Journal (WSJ) [34], noisy な音声コーパスとして Aurora4 [35] を用いた. WSJ 音声コーパスは連続音声認識システムを構築するために収録されたコーパスで, 英語ニュース記事の読み上げ音声で構成されている. WSJ は WSJ0 と WSJ1 の 2 つに分かれており, 合計で 83 時間ほどの音声で構成されている. 一方 Aurora4 は以下の 6 種類の (環境) 雑音を WSJ0 に人工的に付与した音声コーパスである.

- Car
- Crowd of people (babble)
- Restaurant
- Street
- Airport
- Train station

学習データについては上記 6 種類からランダムに選ばれた 1 種類の雑音が 10 dB〜20 dB の範囲でランダムに付与されている. 一方開発データ “dev330” および評価データ “test166” に関しては, 学習データと同様に 6 種類からランダムに選ばれた 1 種類の雑音が 5 dB〜15 dB の範囲でランダムに付与されている. 以上の WSJ0 と Aurora4 の関係により, 今回の実験では WSJ については WSJ0 (15 時間ほどの音声で構成されている) のみを用いた. 使用した発話数の詳細は表 4.5 の通りである.

音響分析には 4.3.1 章の実験と同様に, Kaldi Speech Recognition Toolkit [32] を利用し, 表 4.2 のような条件で音響特徴量を抽出した. 正解ラベルは, アルファベット 26 種類にアポストロフィ, ピリオド, ダッシュ, スペース, noise ラベル, 文終了ラベルの 6 種類の記号を加えた計 32 個のラベルとした.

Bidirectional-LSTM のパラメータや学習条件は表 4.6 の通りである. AT や VAT を適用する上で, 式 (4.2) および式 (4.3) 中のノイズの規模 ϵ を決めるにあたり, 開発データを用いてグリッドサーチを行ない, 最も認識精度が高かったものを用いた. AT については [0.2, 0.3, 0.5] の範囲でグリッドサーチを行い, 0.3 とした時に最も高い精度が得られた. 一方 VAT については [3.0, 5.0, 10.0] の範囲でグリッドサーチを行い, 5.0 とした時に最も高い精度が得られた. 式 (4.2), 式 (4.3)

表 4.7: WSJ0 および Aurora4 における CER

学習データ/モデル	CER(開発データ)	CER(評価データ)
WSJ0	dev93	eval92
CTC [6]	31.23	24.69
CTC	33.30	25.58
CTC + AT	31.28	22.90
CTC + VAT	**29.34**	**21.87**
Aurora4	dev330	test166
CTC	35.71	35.91
CTC + AT	33.67	33.94
CTC + VAT	**33.36**	**33.14**
WSJ0	dev330	test166
CTC	67.05	69.49
CTC + AT	63.86	66.51
CTC + VAT	**62.17**	**64.19**

中の AT および VAT の正則化項の重み α は 1.0 とした.

4.4.2　実験結果

各モデルの認識エラー率は表 4.7 のようになった. WSJ0 や Aurora4 の各コーパスで学習させた結果, CTC 単体のベースラインや先行研究 [6] よりも, AT や VAT による敵対的学習を用いた枠組みの方が低いエラー率が得られた. ベースラインに対しては開発データ・評価データともに 10%以上程度の相対改善率を達成した. この改善は 4.3.1 章の実験でも同様であり, 小規模コーパスにおいても中規模コーパスにおいても精度向上が確認できたことから, 敵対的学習の適用により End-to-end 音声認識の認識率を大きく向上できると結論付けることが可能である. また静音音声コーパスでの学習だけでなく, 雑音音声コーパスの学習についても有効であることが確かめられた.

一方, 敵対的学習による耐雑音性の改善度合いを検証するために, Aurora4 が WSJ0 にノイズを付与して作成されたコーパスであることを利用して, WSJ0 で学習させたモデルに対して Aurora4 の開発データ “dev330” および評価データ “test166” でのエラー率を測定した. すなわち静音音声コーパスのみの学習で雑音音声コーパスに対する精度を測定した. その結果, 同じ条件のもとで学習させたベースラインよりも AT や VAT の方が低いエラー率を達成したものの, この条件下でのエラー率は全般的に悪く, Aurora4 で学習させたベースラインと比較させた場合, 大きく劣る結果となった. 以上から, 敵対的学習の枠組みを導入した場合, 汎化性能は向上するが, 耐雑音性を大きく向上させるまでには至らず, 雑音を付与して人工的に作成したコーパスによる学習がこの枠組みよりも効果的であると言える. これは, 4.3.1 章の実験でも同様だったが, ノイズの探索空間やノイズの加え方が現時点での枠組みでは不適切であるためだと考えられる.

第5章

声道長変換を考慮した敵対的学習の適用の検討

5.1　はじめに

4 章で見たように, End-to-end 音声認識において敵対的学習を適用させることで, clean な音声のみで構成された音声コーパスで学習させた場合でも, noisy な音声を混入させた multi 音声コーパスで学習させた場合でも, どちらの場合でもエラー率を改善させることができることが示されたが, このような枠組みで学習させた場合についても, clean な音声のみを用いて学習させた場合の noisy な音声コーパスに対する認識率は悪いという結果が得られた. これは敵対的学習を適用させる枠組みにおいて, ノイズの加算方法およびノイズの探索空間に問題があると考えられる.

一方, 音声の音響的実態に不可避に混入する非言語特徴は, ケプストラム空間においてアフィン変換の形で表現することが可能である. 特にケプストラム空間における話者の声道長変換は, パラメータ一つでアフィン変換の形で表現することが可能である [36].

そこで, 4 章で行なっていたスペクトルドメインのおけるノイズの加算に加え, 話者の声道長変換を考慮した場合のアフィン変換による特徴量の変換によって敵対的サンプルを生成し, その影響を観測した. これにより, 敵対的学習におけるノイズの加え方を拡張させた場合の影響を観測することが可能になる.

本章では声道長に対する音響特徴量の依存性について説明を行なったのち, 声道長変換を考慮した場合の検討事項について述べ, 音声認識実験における実験を行なう.

5.2　声道長に対する音響特徴量の依存性

5.2.1　非言語的特徴のモデル化

音声に混入する非言語的特徴は主に加算性雑音, 乗算性歪み, 線形変換性歪みの三種類に分類されるとされている [37, 38]. 加算性雑音は時間軸上の加算で表現される雑音であり, 背景雑音がその典型として挙げられる. これらの雑音は場所の移動等の対応が可能であり, 不可避的なものではないといえる. 特に不可避的な歪みとして考えられるのが後者の二つである.

乗算性歪みはスペクトルに対する乗算で表現される歪みであり, ケプストラム空間においては加算演算 $\boldsymbol{c}' = \boldsymbol{c} + \boldsymbol{b}$ で表現することが出来る. この種の雑音の例としては, マイクロフォンの音響特性などが挙げられる. また話者の声道形状差異も一部近似的に乗算性歪みであると考えられる. 音声は必ず発話者を伴い, 音響機器によって収録されるため, これらの歪みは不可避である.

線形変換性歪みはケプストラム空間において行列 $\boldsymbol{A}$ による線形変換 $\boldsymbol{c}' = \boldsymbol{A}\boldsymbol{c}$ で表現される歪みである. スペクトル表現においては, 話者の声道長差異や聴取者の聴覚特性差異は周波数ウォーピングとして考えられる. 周波数ウォーピングはケプストラム空間において線形変換で記述されることが示されている [36]. すなわち声道長差異や聴覚特性差異は近似的に線形変換性歪みとして扱うことができる.

以上をまとめると, 音声の音響的実体に不可避的に混入する非言語的特徴は, ケプストラム空間においてアフィン変換 $\boldsymbol{c}' = \boldsymbol{A}\boldsymbol{c} + \boldsymbol{b}$ で表現される. これらの $\boldsymbol{A}, \boldsymbol{b}$ が話者や収録環境によって多様に変化し, 音声の音響的実体に様々な歪みが混入する事になる. 図 5.1 はアフィン変換 $\boldsymbol{c}' = \boldsymbol{A}\boldsymbol{c} + \boldsymbol{b}$ によって, 音声の対数スペクトルが変化する様子を示したものである. このとき行列 $\boldsymbol{A}$ は周波数方向 (図中の水平方向) への変化となり, 加算するベクトル $\boldsymbol{b}$ は対数パワー方向 (図中の垂直方向) への変化として表れる. 特に線形変換 $\boldsymbol{c}' = \boldsymbol{A}\boldsymbol{c}$ は主に声道長の長さの違いに起因する歪みであり, ケプストラム空間において, ある $\boldsymbol{A}$ による変換が声道長の長さの変化を端的に表すことになる.

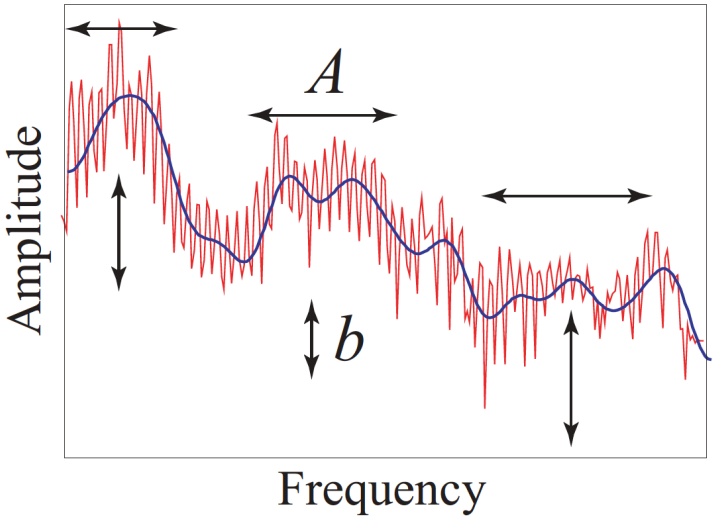


図 5.1: 非言語特徴量によって引き起こされるスペクトル歪み

5.2.2　ケプストラムの声道長依存性

話者の声道長の変化は, 音声のスペクトル表現における周波数ウォーピングとして考えることができる. 今, 周波数ウォーピングにおける変換前後の正規化角周波数を $\omega, \hat{\omega} (0 \leq \omega, \hat{\omega} \leq \pi)$ とする. このとき $z = e^{j\omega}$, $\hat{z} = e^{j\hat{\omega}}$ とし, 周波数ウォーピングとして以下の 1 次全域通過関数を考える.

$$\hat{z}^{-1} = m(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \tag{5.1}$$

このときウォーピングパラメータ α は $|\alpha| < 1$ の実数であり, $\alpha < 0$ の場合, 周波数軸が低域に変換され声道長は長くなる. 一方 $\alpha > 0$ の場合は, 周波数軸は高域に変換され声道長が短くなる. 図 5.2 はウォーピングパラメータを変化させた場合の式 (5.1) の様子である. 以下, 前述のスペクトルドメインにおける周波数ウォーピングをケプストラム空間における記述に置き換える. 江森らは声道長の変化をケプストラム空間で記述し, これらのパラメータ推定に基づく声道長正規化を行っている [39]. パワーを表現するケプストラムの 0 次項 $(c_0, \hat{c}_0)$ を考慮しない場合, 周波数ウォーピングは以下の式で表現される.

$$\hat{\boldsymbol{c}} = \boldsymbol{A}\,\boldsymbol{c} \tag{5.2}$$

$$\hat{\boldsymbol{c}} = (\hat{c}_1\ \hat{c}_2\ \hat{c}_3\ \hat{c}_4 \cdots)^t$$

$$\boldsymbol{A} = \begin{pmatrix} 1-\alpha^2 & 2\alpha - 2\alpha^3 & \cdots & \cdots \\ -\alpha + \alpha^3 & 1 - 4\alpha^2 + 3\alpha^4 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \tag{5.3}$$

$$\boldsymbol{c} = (c_1\ c_2\ c_3\ c_4 \cdots)^t$$

ここで行列 A の要素 a_{ij} は, Pitz らによれば以下のように表せる [36].

$$a_{ij} = \frac{1}{(j-1)!} \sum_{m=\max(0,j-i)}^{j} \binom{j}{m} \times \frac{(m+i-1)!}{(m+i-j)!} (-1)^{(m+i-j)} \alpha^{(2m+i-j)} \tag{5.4}$$

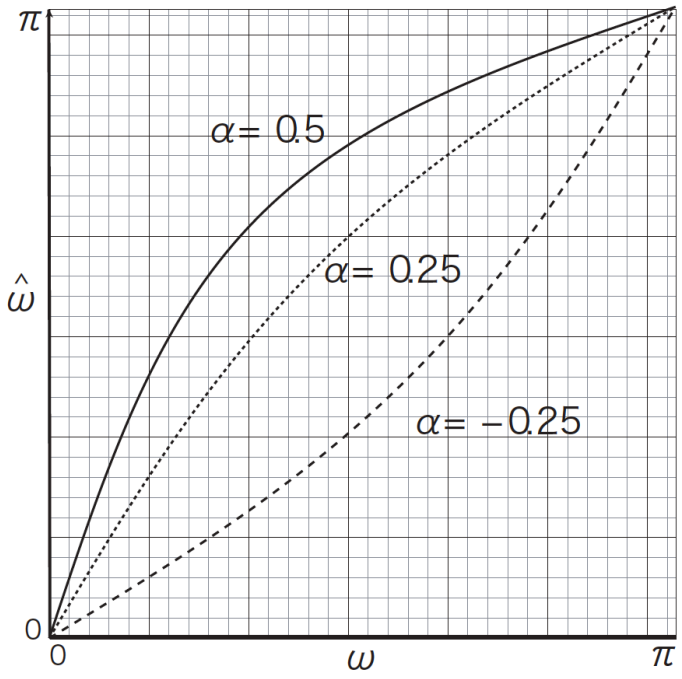


図 5.2: 周波数ウォーピング関数

ただし

$$\binom{j}{m} = \begin{cases} {}_jC_m & (j \geq m) \\ 0 & (j < m) \end{cases} \tag{5.5}$$

となる．$\alpha \ll 1$ の場合には 2 次以上を無視できるので, 式 (5.3) は

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2\alpha & 0 & \cdots \\ -\alpha & 1 & 3\alpha & \cdots \\ 0 & -2\alpha & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \tag{5.6}$$

と近似できる．また式 (5.3) の変換行列は強い回転性を有し, その回転性には, 特に話者の声道長の違いが大きくなる場合には音韻や性別などによる影響が見られることが指摘されている [40].

5.3　敵対的学習の適用における声道長変換の考慮

4 章では, 入力特徴量 $\boldsymbol{x}$ に対するノイズ $\boldsymbol{r}$ の付与を $\boldsymbol{x}+\boldsymbol{r}$ で表されるようなスペクトルドメインにおける加算で表現していたが, これを $\boldsymbol{A}\boldsymbol{x}+\boldsymbol{r}$ のようなアフィン変換に置き換えることを考える．これにより線形変換性歪みを考慮することができるようになる．今回は, 変換行列 $\boldsymbol{A}$ を式 (5.3) のように仮定することで, 話者の声道長変換を考慮した場合のアフィン変換による特徴量の変換によって敵対的サンプルを生成し, その影響を観測した．これにより, 敵対的学習におけるノイズの加え方を拡張させた場合の影響を観測することが可能になる．またこの方法はウォーピングパ

表 5.1: 音響分析条件

項目	設定値
サンプリング	16 bit / 16 kHz
窓	25 ms 幅 / 10 ms シフト
音響特徴	(i) FBANK: 対数メルフィルタバンク (40 次元) + Δ・ΔΔ 特徴量 (計 120 次元)
	(ii) MFCC: MFCC (40 次元) + Δ・ΔΔ 特徴量 (計 120 次元)

ラメータ α のみを設定すれば変換行列 $\boldsymbol{A}$ が決まるため話者の声道長変換を簡便に考慮することが可能である.

具体的な手順としては以下のようになる.

1. 発話ごとにウォーピングパラメータ α をランダムな値で設定する. ここで $\alpha \ll 1$ を満たすように設定することで, 変換行列 $\boldsymbol{A}$ を式 (5.6) のように近似を行なうことができ, 簡便に変換行列を導出することができる.
2. AT および VAT の枠組みにより, 敵対的な摂動 $\boldsymbol{r}$ を毎フレーム求める.
3. サンプル $\boldsymbol{x}$ に対して敵対的サンプル $\boldsymbol{Ax} + \boldsymbol{r}$ を生成し, 目的関数にしたがって学習を行う.

以上のようにして, 敵対的学習のノイズの加え方を拡張させた場合の影響を観測することを試みる.

5.4　評価実験

5.4.1　実験設定

音声コーパスには WSJ0 および Aurora4 を利用した. データセットの内訳は表 4.5 の通りである. 今回の実験では, 学習には WSJ0 のみ利用し, 主に雑音有コーパス Aurora4 で評価を行った場合の CER の変化を観測した. 音響分析には Kaldi Speech Recognition Toolkit [32] を利用し, 表 5.1 のような条件で音響特徴量を抽出した. 今回メルフィルタバンクだけでなく, ケプストラム特徴量である MFCC も用いたのは, 前述の声道長変換がケプストラム空間における記述であるためである. 正解ラベルは, アルファベット 26 種類にアポストロフィ, ピリオド, ダッシュ, スペース, noise ラベル, 文終了ラベルの 6 種類の記号を加えた計 32 個のラベルとした.

End-to-end 音声認識を行なう Bidirectional-LSTM のパラメータや学習条件は表 4.6 と同様である. また AT におけるノイズの規模 ϵ は 0.3, VAT におけるノイズの規模 ϵ は 5.0 とした. 目的関数に置ける AT および VAT の正則化項の重み α は 1.0 とした. ウォーピングパラメータ α は平均 0, 分散 0.05, 絶対値 1 未満の切断正規分布に従ってランダムに決定させ, 発話ごとに固定した.

5.4.2　実験結果

声道長変換を仮定したアフィン変換を行なった場合の認識エラー率は表 5.2 のようになった. この表において, (Affine) と追加されている手法が, 声道長変換を考慮して敵対的学習による学習を行なったモデルである. FBANK, MFCC どちらの特徴量を用いて学習を行った場合でも, 声道長変換を考慮することによってエラー率の改善がみられた. この傾向は AT・VAT どちらにおいても見られたため, 敵対的学習におけるアフィン変換の枠組みは有効であるということが示された. しかしながら, 4.4 章での実験と同様, 雑音無コーパスのみを利用し敵対的学習の枠組みでノイズ

表 5.2: 声道長変換を考慮した場合の Aurora4 データセットに対する CER

特徴量	学習コーパス	手法	開発データの CER	評価データの CER
FBANK	Aurora4	CTC	35.71	35.91
	WSJ0	CTC	67.05	69.49
		CTC+AT	63.86	66.51
		CTC+AT(Affine)	61.72	62.91
		CTC+VAT	62.17	64.19
		CTC+VAT(Affine)	60.81	60.96
MFCC	Aurora4	CTC	36.54	36.43
	WSJ0	CTC	68.63	68.83
		CTC+AT	64.19	67.73
		CTC+AT(Affine)	61.91	61.66
		CTC+VAT	64.81	66.19
		CTC+VAT(Affine)	61.38	61.71

を付与して学習を行なったモデルと, 雑音有コーパスで学習を行なったモデルとでエラー率を比較した場合は, 前者のエラー率は 4.4 章での実験よりも改善が見られたものの, 後者には及ばない結果となった.

次に, 特徴量を変化させた場合についての議論を行なう. 今回のアフィン変換では前述のような声道長変換を仮定しており, この声道長変換がケプストラム空間における記述であったため, ケプストラム特徴量の一つである MFCC を用いた場合についても実験を行なった. FBANK と MFCC について比較を行なうと, アフィン変換を行なわないモデルに関しては, FBANK の方が総じて高い精度が得られた. これは 2.2.3 章で述べたように, 特徴量抽出の範囲まで DNN に担わせることが出来る FBANK では特徴量抽出の最適化まで行なうことができるため, FBANK が MFCC よりも優ったのだと考えられる. 一方, 声道長変換の処理の影響は MFCC の方が FBANK よりも大きく, より大きな改善率を達成する結果となった. これは前述したように, 声道長変換がケプストラム空間における記述であったため, ケプストラム特徴量を用いた場合の方が影響が大きいためだと考えられる. ただし FBANK の場合にもアフィン変換による効果が見られていることから, DNN の内部で擬似的に FBANK からケプストラム特徴量に変換する処理が行なわれているということが考えられる.

今回の枠組みでは声道長変換を明に仮定したアフィン変換を同時に行ない, その結果エラー率の改善が見られたため, アフィン変換を行なう変換行列についても敵対的学習の枠組みのように, モデルが誤識別をするような変換行列を推定する枠組みを導入することが出来れば, 声道長変換だけでなく柔軟に変換行列を表現できるようになるため, よりエラー率が改善されると期待できる.

第6章

結論

6.1　本研究のまとめ

本研究では End-to-end 音声認識に敵対的学習, 特に adversarial training および virtual adversarial training を適用させることで, 耐雑音性をはじめとした汎化性能の向上を目指した. 敵対的学習は汎化性能を向上させる手法の一つであるが, 特に敵対的サンプルを生成する枠組みは音声に雑音を付与する処理とみなすことができ, この手法によって付与される雑音は種類を全く仮定する必要がなく, 学習の各ステップで毎度自動で算出されるため, 耐雑音性の向上に繋がることが期待される. またこの枠組みでは学習データの明示的な増量が不要であるため, ネットワークの最適化の難化といった問題を避けることが可能である. TIMIT を対象とした音素認識実験では静音環境下および雑音環境下ともに精度改善が確認された. 一方, WSJ0 や Aurora4 といった 中規模なデータセットで音声認識実験では, それぞれのコーパスでの学習では精度向上が確認でき, 汎化性能を向上させることに成功したが, 耐雑音性に関しては, 学習コーパスに雑音を付与する方法と比較すると大きく精度が劣っていたため, 単純に敵対的学習を適用しても有効でないことが判明した. また, 敵対的学習の枠組みで生成したノイズが付与された特徴量を可視化させたが, ノイズ付与前に対してほとんど差が見られず, 現実のノイズ付与と大きくことなる結果が得られた.

以上を踏まえ, ノイズの加え方に課題があると考えられるため, 話者の声道長変換を考慮した場合のアフィン変換による特徴量の変換によって敵対的サンプルを生成し, 音声認識実験でその影響を観測した. 学習コーパスに雑音を付与する方法には依然として耐雑音性の面で及ばないが, ノイズの加え方を修正したことによってエラー率の改善が見られた. 今回導入した敵対的学習の枠組みをノイズの探索空間およびノイズの加え方の面で更に拡張することで, 耐雑音性の向上が達成可能だと考えられる.

6.2　今後の課題

敵対的学習の枠組みにおいて, ノイズの探索範囲およびノイズの加算方法に改善の余地がある. 前者について, 例えば特徴量から話者性の影響を取り除いた上で探索を行なうことで, よりモデルの学習に有効なノイズを生成することが可能になる. このように, 同じ正解ラベルが割り当てられているのに特徴量を変化させてしまうような要因を排除するのが有効な手法だと考えられる. 後者については, 今回の枠組みで生成され付与されたノイズが乗算性のノイズであると予想されることから, 加算性のノイズにも乗算性のノイズにも, 更には線形変換性のノイズにも対応出来るようノイズの加え方をより一般的な式で表記し, モデルを最も誤認識させるようにそれぞれの成分のノイズを求めるような敵対的学習の枠組みに落とし込むのが有効であると考えられる. その際, 今回は入力に音響特徴量を用いたが, 生の音声をそのままネットワークに入力させて十分な認識を行なうことができれば, 今回の枠組みで生成されたノイズを音で実際に聞くことが可能になり, より効果的な手法の糸口になることが期待される.

謝辞

本研究を進めるにあたり多くの方にお世話になりました.

指導教官である峯松信明教授には 2 年間に渡り研究や発表, 論文執筆など手厚いご指導を賜りましたこと, 深く感謝いたします. 特に修士 2 年の時に書いた英語論文では, 常日頃からお忙しい中, 内容だけでなく細かい文法ミスやおかしな言い回しなど細部まで何度も丁寧に添削していただきました. 締め切りぎりぎりなことが多く大変ご迷惑をおかけいたしましたが, 非常に勉強になりました. また, TA やチューターとしての仕事を与えていただいたことも大学院生活の励みになりました. 日頃の研究活動を支えて下さった高橋登技術専門員, 池上恵事務補佐員にも感謝いたします.

齋藤大輔講師には, 理論の構築や実装に関する知識, 研究方針など, 非常に多くの相談に乗っていただきました. 研究環境についても常日頃からお気遣い頂き, 円滑に研究を進めることができました. 大変お世話になりました. 研究室の皆様には, お互いの研究の議論や雑談を行なったり, ご飯に付き合っていただいたりと, 非常に楽しい時間を過ごさせていただきました. 充実した 2 年間を過ごすことができたのは皆様のおかげです.

最後に, 今まで長いこと学生であることを認め支えてくれた家族に感謝いたします. ありがとうございました.

2018 年 2 月 1 日
増田 嵩志

参考文献

[1] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning*, 2014, vol. 32, pp. 1764–1772.

[2] Geoffrey E. Hinton, Li Deng, Dong Yu, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 369–376.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, 2015.

[5] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 4, pp. 745–777, 2014.

[6] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4835–4839, 2017.

[7] Dario Amodei, Rishita Anubhai, Eric Battenberg, *et al.*, "Deep speech 2 : End-to-end speech recognition in english and mandarin," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, vol. 48, pp. 173–182.

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[9] Takeru Miyato, Shinichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii, "Distributional smoothing with virtual adversarial training," in *International Conference on Learning Representations*, 2016.

[10] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[11] Mark Gales and Steve Young, "The application of hidden markov models in speech recognition," *Found. Trends Signal Process.*, vol. 1, no. 3, pp. 195–304, 2007.

[12] Lawrence Rabiner and Biing-Hwang Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Inc., 1993.

[13] Frank Seide, Gang Li, and Dong Yu, "Conversational speech transcription using context-dependent deep neural networks," in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association*, 2011, pp. 437–440.

[14] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, 2010, pp. 1045–1048.

[15] 河原 達也, "音声対話システムの進化と淘汰 ー歴史と最近の技術動向ー," **人工知能学会誌**, vol. 28, no. 1, pp. 45–51, 2013.

[16] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, Yifan Gong, Alex Acero, and Mike Seltzer, "Recent advances in deep learning for speech research at microsoft," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8604–8608.

[17] Daniel Jurafsky and James H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall PTR, 2000.

[18] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 2048–2057.

[19] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "EESEN: end-to-end speech recognition using deep RNN models and wfst-based decoding," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2015, pp. 167–174.

[20] Ying Zhang, Mohammad Pezeshki, Philémon Brakel, Saizheng Zhang, César Laurent, Yoshua Bengio, and Aaron Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," in *INTERSPEECH 2016, 17th Annual Conference of the International Speech Communication Association*, 2016, pp. 410–414.

[21] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015, pp. 577–585.

[22] Liang Lu, Xingxing Zhang, KyungHyun Cho, and Steve Renals, "A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association*, 2015, pp. 3249–3253.

[23] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 4945–4949.

[24] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 4960–4964.

[25] Liang Lu, Xingxing Zhang, and Steve Renals, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 5060–5064.

[26] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] M. Schuster and K.K. Paliwal, "Bidirectional recurrent neural networks," *Trans. Sig. Proc.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[28] Ilya Sutskever, Oriol Vinyals, and Quoc V. V Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, pp. 3104–3112. Curran Associates, Inc., 2014.

[29] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," in *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*, 2017, pp. 949–953.

[30] Rohit Prabhavalkar, Kanishka Rao, Tara Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*, 2017, pp. 939–943.

[31] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon Technical Report N*, vol. 93, 1993.

[32] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011, pp. 1–4.

[33] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2014.

[34] Doug Paul John Garofalo, David Graff and David Pallett, "CSR-I (WSJ0) Complete," Linguistic Data Consortium, Philadelphia, USA, 2007.

[35] N Parihar and J Picone, "Aurora working group: DSR front end LVCSR evaluation AU/384/02," *Inst. for Signal and Information Process, Mississippi State University, Tech. Rep*, 2002.

[36] Michael Pitz and Hermann Ney, "Vocal tract normalization equals linear transformation in cepstral space," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 5, pp. 930–944, 2005.

[37] 峯松 信明, 西村 多寿子, 西成 活裕, and 櫻庭 京子, "構造不変の定理とそれに基づく音声ゲシュタルトの導出," **電子情報通信学会技術研究報告**, vol. 105, no. 98, pp. 1–8, 2005.

[38] Nobuaki Minematsu, "Mathematical evidence of the acoustic universal structure in speech," in *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005, pp. 889–892.

[39] Tadashi Emori and Koichi Shinoda, "Rapid vocal tract length normalization usgin maximum likelihood estimation," in *Eurospeech 2001, 7th European Conference on Speech Communication and Technology*, 2001, pp. 1649–1652.

[40] Daisuke Saito, Nobuaki Minematsu, and Keikichi Hirose, "Rotational properties of vocal tract length difference in cepstral space," *Journal of ResearchInstitute of Signal Processing*, vol. 15, no. 5, pp. 363–374, 2011.

発表文献

国内研究会・全国大会

[1] 増田嵩志, 張豪逸, 磯健一, “LSTM を用いたキーワードスポッティング” 日本音響学会春季講演論文集, pp. 177–178, 2017.

[2] 増田嵩志, 齋藤大輔, 峯松信明, “耐雑音性の向上を目的とした End-to-end 音声認識における Virtual Adversarial Training の適用” 日本音響学会秋季講演論文集, pp. 7–10, 2017. (**学生優秀発表賞**)

[3] 増田嵩志, 齋藤大輔, 峯松信明, “敵対的学習を適用した End-to-end 音声認識” 情報処理学会音声言語情報処理研究会資料, pp. 1–5, 2017.

学位論文

[4] 増田嵩志, “ニューラルネットワークを利用した日英機械翻訳における品詞情報の利用” 東京大学工学部電子情報工学科卒業論文, 2016