NAIST-IS-MT1651110

修士論文

End-to-End 音声認識モデルの圧縮

森 巧磨

2018 年 3 月 15 日

奈良先端科学技術大学院大学
情報科学研究科 情報科学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士（工学）授与の要件として提出した修士論文である.

森 巧磨

審査委員：

中村 哲 教授	（主指導教官）
松本 裕治 教授	（副指導教官）
Sakriani Sakti 特任准教授	（副指導教官）

End-to-End 音声認識モデルの圧縮*

森 巧磨

内容梗概

大規模な音声コーパスが作成され，音声認識（ASR）のモデルとして，大規模なニューラルネットワークが使用されるようになった．また，シンプルな手法として，殆どのモジュールを単一のモデルとして学習する End-to-End 音声認識の研究が行われている．ニューラルネットワークを用いた音声認識は，たくさんのパラメータを有し，新しいデータの学習および予測のために多くの計算資源を必要とする．パラメータが増える原因の一つとして，時系列タスクをモデリングし，様々な複雑な問題を解析する手法としてリカレントニューラルネットワーク（RNN）を使うことがあげられる．音声認識は音声要約，自動コールセンター，音声翻訳などの多くのアプリケーションの重要なコンポーネントである．したがって，多くの場面でこのモデルを使えるようにするためには，メモリを削減して，軽量なモデルとする必要がある．本稿では，End-to-End 音声認識を 2 つのアプローチで圧縮することを試みた．一つは，知識蒸留に基づいた方法である．異なる出力長を同様に学習できる損失関数 Connectionist Temporal Classification（CTC）を利用し，巨大な教師ネットワークの出力文を小さな生徒ネットワークで訓練して，性能向上を狙う．もう一つは，リカレントネットワークの中間層をテンソル表現し，それらを効率的に分解するテンソルトレイン分解により，パラメータ数を大幅に削減する代替 RNN モデルを提案する．我々は，音声コーパスである Libri Speech において非圧縮ゲーティッドリカレントユニット（GRU）モデルとテンソルトレイン分解により圧縮した GRU モデルを比較評価し，パラメータの数を大幅に削減しながら性能を維持することを示した．

キーワード

*奈良先端科学技術大学院大学 情報科学研究科 情報科学専攻 修士論文, NAIST-IS-MT1651110, 2018 年 3 月 15 日.

テンソルトレイン分解，モデル圧縮，End-to-End モデル，音声認識

End-to-End Automatic Speech Recognition Model Compression*

Mori Takuma

Abstract

A large-scale speech corpus was created, and a large-scale neural network was used as a model of Automatic Speech Recognition (ASR). As a simple method, research on End-to-End ASR that trains most modules as a single model is being conducted. ASR using neural networks has many parameters and requires a lot of computational resources for training and prediction of new data. One of the reasons for increasing the number of parameters is to use Recurrent Neural Networks (RNN) as a method of modeling time series tasks and analyzing various complicated problems. ASR is an important component of many applications such as speech summarization, call center, speech translation and so on. Therefore, in order to be able to use this model in many scenes, it is necessary to make it a lightweight model. In this paper, we tried to compress End-to-End ASR model with two approaches. One is a method based on Knowledge Distillation. Using the fact that Connectionist Temporal Classification (CTC) can learn different output lengths, in the same way, we aim to improve performance by training output sentences of a huge teacher network with a small student network. The other approach is to an alternative RNN model that drastically reduces the number of parameters of the middle layers of recurrent network by tensor decomposition. The Gated Recurrent Unit (GRU) model compressed by Tensor Train Decomposition and the uncompressed GRU model were compared and evaluated by the speech corpus Libri Speech. We showed that tensor train decomposition maintains performance while greatly reducing the number

*Master's Thesis, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1651110, March 15, 2018.

of parameters.

Keywords:

Tensor Train Decomposition, Model Compression, End-to-End Model, Automatic Speech Recognition

目次

図目次

表目次

第 1 章　緒言

1.1　背景

音声認識 (Automatic Speech Recognition, ASR) は入力信号を音声特徴ベクトルに変換し, その音声特徴ベクトルの系列から対応する単語列を推定することである．1952 年に Bell らが単一話者の数字認識システムを構築したことから始まっている [1]．音声認識は統計的モデルに基づく手法の確立や大規模コーパスの作成，計算機の高性能化によって，大きく発展し，様々なアプリケーションで利用されている [2]．身近な例では，スマートフォンに搭載されている音声検索システム [3, 4] や音声入力システムにより操作可能なカーナビゲーションシステム，国会の会議録作成にも音声認識が使用されている [5, 6]．音声認識は，その問題の複雑さから，音声から直接テキストを推定することは難しく，様々なヒューリスティクスを用いた特徴抽出といくつかのモジュールを用いながら段階的に学習されていた．しかし，最近では音声から直接テキストを推定する流れができている．まず，計算機の高性能化といくつかの最適化に関する発見により，深い多層ニューラルネットワークを用いて，複雑なパターン認識を解くことが可能になった．また，大規模な音声コーパスが作成されたことにより，殆どのモジュールを単一のモデルとして音声から直接テキストを予測する End-to-End 音声認識の研究が可能になった [7, 8, 9].

1.2　研究目的

深層学習を用いた音声認識モデルは，複雑な問題を解決するため，非常に膨大なパラメータを使う必要があり，入力音声を認識するために高性能なサーバを必要とする．音声認識は様々な使用環境があり，モバイルデータ接続が利用できない場合や，通信が遅く，信頼性が低い場合でも，ユーザーのやりとりを可能にしながら，待ち時間を短縮する必要がある．これらの主な問題は，デバイスのディスク，メモリ，および計算量の制約である．ニューラルネットワークの計算量はモデルパラメータの数に比例するため，モデルのパラメータを圧縮することは，計算量とメモリ消費量を削減する観点から不可欠である．そこで，本稿では図 1.1 のように精度をなる

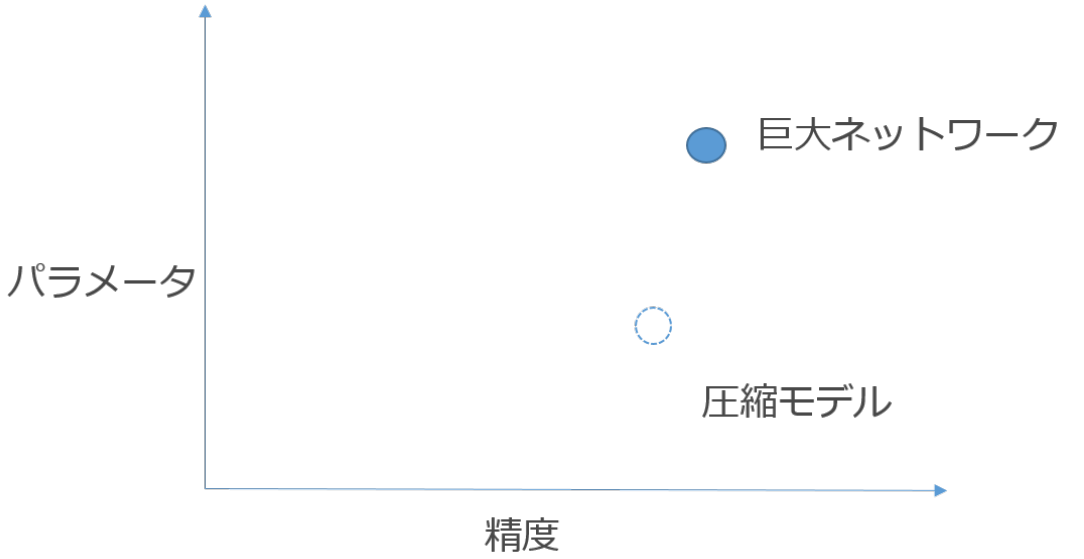


図 1.1　本研究の目標

べく維持しつつパラメータを削減する 2 つの End-to-End 音声認識のモデル圧縮を提案する．

1.3　関連研究

ニューラルネットワークのモデル圧縮に関する研究は大きく分けると四つのアプローチで行われている．

一つ目は，ニューラルネットワークのパラメータの冗長性に着目した研究である．Denil らは，ニューラルネットワークの一部のパラメータだけを学習し，残りのパラメータを推定することで，パラメータの削減を出来ることを示している [10]．また，Han らは，ニューラルネットのパラメータの一部を 0 にして，再学習することで，パラメータ数を削減しながら性能を維持できる Pruning を提案した [11, 12]．また，Han らはパラメータを k-means 法でクラスタリングし，似ているパラメータを共通のパラメータにする Weight Sharing でパラメータの削減を行えることを示した [12]．

二つ目は，ニューラルネットワークのパラメータが行列で表現されていること

に着目した研究である．Sainath らは最終層のパラメータを低ランク行列分解することにより，パラメータを削減できることを示した [13]．Denton らは，畳み込みニューラルネットのフィルタを特異値分解することにより，パラメータの削減と高速化を行えることを示した [14]．Wang らは音響モデルを圧縮するために特異値分解（SVD）とベクトル量子化の組み合わせて全結合層を圧縮することに成功した [15]．Rohit らは RNN(Recurrent Neural Network) の回帰入力と入力のパラメータを因数分解することにより圧縮できることを示した [16]．また，Alexander らは VGG(Simonyan らが ImageNet Large Scale Visual Recognition Challenge 2014[17] で提案したモデル [18]) の全結合層のパラメータをテンソルトレイン分解で圧縮できることを示した [19]．また，Andros らは MIDI 音楽分類で RNN や GRU（Gated Recurrent Unit）のパラメータをテンソルトレイン分解圧縮できることを示した [20].

三つ目は，ニューラルネットワークのパラメータに用いられる浮動小数点型の冗長性に着目した研究である．Lin らは畳み込みニューラルネット層で用いている浮動小数点型を通常の 32bit から 8bit に変更しても殆ど性能が変わらないことを示した．[21]

四つ目は，学習させたいニューラルネットワークより高性能なモデルの学習結果をヒントに学習させる研究である．Bucilua らは，いくつかのニューラルネットワークを正解ラベル付きデータで学習した後，これらのアンサンブルモデルで正解ラベルのないデータにラベルを付与し，拡張したデータで単一モデルの学習をすることで，元のアンサンブルモデルより良い性能を示す，Mimic Network を提案した [22]．また，Hinton らは，統計的な温度で複数のモデルの出力の加重平均を調整しながら，アンサンブル学習を単一のモデルに圧縮する方法を提案した [23]．Hinton らはこのような複数のモデルや高性能なモデルの出力分布を学習のターゲットとし，単一の小さなモデルを学習させ，高い性能を得ることを知識蒸留と呼んでいる [23]．また，Bulo らは，多くのドロップアウトパターンを作り，異なる問題に対して強いモデルを作り，これを知識蒸留し，単一の高性能なモデルを作った [24]．機械翻訳の分野では，Kim らは時系列学習のための Sequence-to-Sequence 注意モデルの知識蒸留を提案し，教師ネットワークの系列出力分布や出力文字列を学習する方法などを提案した [25].

本稿では，組み合わせ可能な 2 つの圧縮モデルを提案する．一つは，知識蒸留に基づいた方法である．CTC（Connectionist Temporal Classification）が異なる出力長を同様に学習できることを利用し，巨大な教師モデルの出力文を小さな生徒モデルで訓練して，性能差の解消を狙う．もう一つは，テンソルトレイン分解を用いて，できるだけ小さなモデルから性能を導出することを検討する．テンソルトレイン分解とは，高次元テンソルを低次元テンソルに圧縮する手法であり，空間/時間計算量ともにテンソルの階数を指数に持たないため，高次元テンソルであっても高速に計算することができる．

1.4 章構成

本稿では，End-to-End 音声認識の 2 つの圧縮方法を検討する．本稿では，End-to-End 音声認識に焦点を当てているが，この提案は，手書き認識や機械翻訳などの他の分野の難しいタスクにも適用できる．第 2 章では，音声認識やニューラルネットワークの基礎的な知識について整理する．第 3 章では，知識蒸留に基づいた提案を行う．第 4 章では，テンソルトレイン分解に基づいた提案を行う．提案した方法の有効性は第 5 章で検証する．最後に，第 6 章にて結果をまとめる．

第 2 章　音声認識と深層学習

本章では，2.1 節において，音声認識システムの概要について述べる．2.2 節において，深層学習の概要について述べる．2.3 節において，本章のまとめを行う．

2.1　音声認識システム

音声認識 (Automatic Speech Recognition, ASR) は入力信号を音声特徴ベクトルに変換し, その音声特徴ベクトルの系列から対応する単語列を推定することである．従来，音声認識は隠れマルコフモデル (Hidden Markov Model, HMM) を用いたモデルが用いられていた．近年，深層学習を用いた End-to-End 自動音声認識モデルという音声から対応する単語列を推定する研究が登場している．

2.1.1　隠れマルコフモデルを用いた音声認識

入力音声から周波数分析により音声特徴ベクトル X を求めるのが音声分析である. 音声分析により得られた音声特徴ベクトル X を用い，音素などの単位で標準的な特徴量パターンを保持しておき, 入力と照合して尤度 $p(X/W)$ を与えるものを音響モデルという．一般的に出力分布には混合正規分布 (Gaussian mixture model, GMM) が用いられ，left-to-right 隠れマルコフモデル (Hidden Markov Model, HMM) を用いてモデル化する．隠れマルコフモデルを用いた音声認識モデルの学習では，音素に対して最適化を行うが，書き起こし文には音素での表記は行わない．そこで，単語辞書を用いる．単語辞書は認識対象の語彙と音素系列を対応づけたもので, この単語辞書により，音素列から単語列に変換される. また，音響モデルとは別に，言語モデルと呼ばれる単語の連鎖に関する制約や尤度 $p(W)$ を規定するモデルの学習も行う．言語モデルでは一般的に n-gram モデルなどを用いる. 最終的には，これらのモデルを統合して，入力音声に対して最尤の単語列を得る. これを，GMM-HMM 音声認識と呼ぶ [26]．Bourlard らは HMM の各状態の出力確率分布として,GMM の代わりにニューラルネットを用いる方法を提案をした (DNN-HMM)[27]．このニューラルネットの入力層は入力特徴ベクトルの次元と同

じユニット数，或いは前後のフレームの特徴ベクトルも併せて入力される. 出力層は HMM のすべての状態数と同じ数のユニットをもち, 各々のユニットが 1 つの状態に対応している. このニューラルネットの学習の手順においては, まず,GMM を出力確率分布として用いた HMM(GMM-HMM) とビタービアルゴリズムをによって，音声データの各フレームとのアラインメントをとる．これを用いて，各フレームに対する音素の one-hot ベクトルを作る．この教師信号を用いてニューラルネットの学習を行う．この提案は，最初はそれほどインパクトのあるものではなかったが，計算機の性能の向上や，大規模コーパスの作成，ニューラルネットの良い初期値の探索方法が見つかることにより，GMM-HMM のモデルを圧倒するようになる [28].

2.1.2 End-to-End 音声認識

これまでに説明した，従来の音声認識システムでは, 音声分析, 音響モデル, 言語モデル, デコーダー, 発音辞書などの多くの要素から構成されていた.DNN-HMM に至っては，2 種類の音響モデルを学習する必要がある．つまり，音声認識はこれらの要素の研究の組み合わせて実現されるものである. 音声認識の性能向上のためには, それぞれのモジュールをどのように設計するかが重要であった. しかし, モジュール毎の誤識別を最小とする規準がないため，様々な組み合わせを行い，性能評価をする必要があった．これに対して，深層学習によって, いくつかのモジュールを単一のモデルで表現する研究が行われている．例えば, 音響モデルの学習は単一のモデルで学習し，音声特徴から音素列を直接予測する研究がある [29, 30]．これにより，また，音声特徴から文字列を直接学習する研究も行われており [8, 9]，これにより発音辞書が不要となった．これらのように，複数の手続きをつなげて 1 つの手続きとして学習することを End-to-End 学習とよばれている.End-to-End 音声認識は，モデルの途中にヒューリスティクスが入らないため，学習が難しいが，多くのデータを使うことにより，人が聴きとった場合の単語誤り率に匹敵することが示されている [9].

2.2 深層学習

深層学習 (Deep Learning) とは, 深く, 多くの層を持ったニューラルネットワークモデルを用いた機械学習の総称である. 近年, 大量のデータと, 強力な分散並列計算を基盤とした深層学習が, 音声認識や一般物体認識などのタスクで高い性能を示したことなどを背景として注目され, 盛んに研究されているのである.

2.2.1 全結合型ニューラルネットワーク

全結合型ニューラルネットワーク (Fully-Connected Neural Network, FCNN) とは層状に並べたユニットが隣接層間でのみ結合した構造を持ち，情報が入力側から出力側に一方向にのみ伝播するニューラルネットワークである．全結合型ニューラルネットワークは以下のように定式化を行う．

$$h = \sigma_h(W_h x + b_h) \tag{2.2.1}$$
$$y = \sigma_y(W_y h + b_y) \tag{2.2.2}$$

このとき，x は入力，h は隠れ層から隠れ層への入力，y は出力である．W, U は重み行列で b はバイアスである．σ_h と σ_y 活性化関数である．

2.2.2 リカレントニューラルネットワーク

リカレントニューラルネットワーク (Recurrent Neural Network, RNN) とはノードの出力が次の時刻における自身のノードの入力になっているようなニューラルネットの総称であり, 入力や隠れ層出力の時間的な依存性を直接モデル化したニューラルネットであるといえる. ニューラルネットワークに時系列に回帰する閉路をつけたものであり，有名なものにエルマンネットワーク [31] とジョーダンネットワーク [32] があるが，音声認識で使われているのはエルマンネットワークである．エルマンネットワークは以下のように定式化を行う．

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \tag{2.2.3}$$
$$y_t \quad = \sigma_y(W_y h_t + b_y) \tag{2.2.4}$$

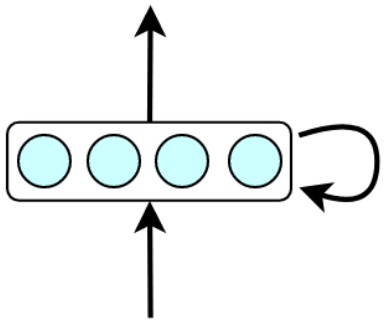


図 2.1　リカレントニューラルネットワークの構造

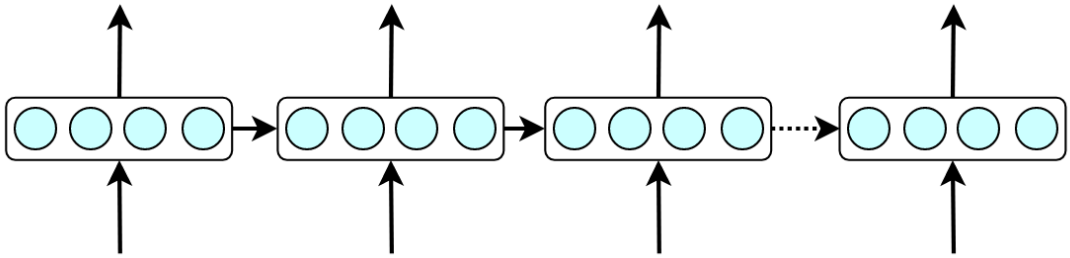


図 2.2　リカレントニューラルネットワークの時間方向への展開

このとき，x_t は入力，h_t は隠れ層から隠れ層への入力，y_t は出力である．W, U は重み行列で b はバイアスである．σ_h と σ_y 活性化関数である．

2.2.3　Gated Recurrent Unit

RNN の Backpropagation through time 学習は時間方向に深いニューラルネットと入出力方向に浅いニューラルネットを結合し，パラメータを共有している．このような深いネットワークでは，誤差逆伝播法における誤差信号が意図した大きさで伝播せず，隠れ層を経るごとに勾配が小さくなってしまう勾配消失問題を抱えている [33]．このため，RNN はモデルの設計上では無限長の記憶を保持することができるようになっているが，実際には，時間方向の学習を上手く行うことが難しいという欠点がある．ニューラルネットの勾配消失問題は，誤差逆伝播法の過程にお

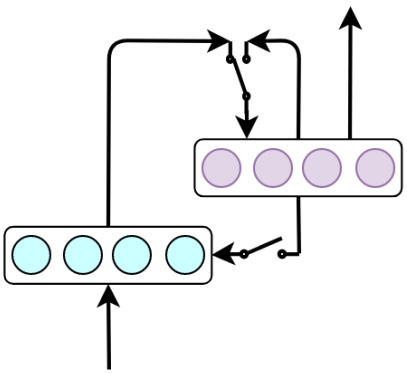


図 2.3　Gated Recurrent Unit の構造

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．活性化関数の微係数が 1 よりも小さい場合は，パラメータの勾配は層を経るごとに指数的に小さく収縮し，また，1 よりも大きい場合は層を経るごとに指数的に拡大する. どちらの場合にしても，勾配のスケールに大きなばらつきを生み，結果として勾配消失問題は勾配法による最適化を困難にする. これの解決を図ったモデルの一つが Gated Recurrent Unit(GRU) である．GRU の定式化は以下のように行う．

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{2.2.5}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{2.2.6}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{2.2.7}$$

このとき，x_t は入力ベクトル，h_t は出力ベクトル，z_t は更新ゲートベクトル，r_t はリセットゲートベクトルである．W, U は重み行列で，b はバイアスである．σ_g はシグモイド関数，σ_h は活性化関数である．

Gated Recurrent Unit は上の式が示すように，更新ゲートとリセットゲートを設置したことによって，勾配消失を防ぐことを狙っている．リセットゲートでは 1 時刻前の出力を忘却する割合を制御している．更新ゲートでは，どの程度情報を保持するかの制御を行っている．

2.2.4　畳み込みニューラルネットワーク

畳み込みニューラルネットワーク (Convolutional Neural Networkg; CNN) は周波数解析などで用いられる畳込み演算を導入したニューラルネットである．複数チャンネルの行列が入力となっており，出力も同様である．

1982 年，Fukushima は add-if silent 則で最適化する畳み込みネットワークであるネオコグニトロンが手書き文字認識に有効であることを提案 [34] し，1989 年には Waibel らは誤差逆伝搬で最適化するニューラルネットで畳み込み構造を持つニューラルネットワークである Time Delay Neural Network が音声認識に有効であることを示した [35]．また，同年，LeCun らは現在では LeNet-5 と呼ばれる手書き文字認識で高い性能を示す畳み込みニューラルネットワークを提案した [36]．そして，2012 年の ImageNet Large Scale Visual Recognition Competition(ILSVRC)[37] という画像認識タスクのコンペティションで，Krizhevsky らが従来の他の統計的学習手法に大きな差をつけたことで注目を集め初めた [38]．また，音声認識タスクでも，音声波形をスペクトログラムに変換し，それを画像として扱うことにより適用し，高い成果を収めている [39, 40, 8]．

本研究では，この畳み込みニューラルネットワークの中でも 1 次元の畳み込みを行う 1D CNN を用いている．1D CNN は以下の式で定式化される．

$$\mathrm{out}(N_i, C_{out_j}) = \mathrm{bias}(C_{out_j}) + \sum_{k=0}^{C_{in}-1} \mathrm{weight}(C_{out_j}, k) \star \mathrm{input}(N_i, k) \tag{2.2.8}$$

このとき，N はバッチサイズ，C はチャンネル数，L は入力長である．

2.3　まとめ

本章では，2.1 節において，音声認識システムの概要，2.2 節において，深層学習の概要について述べた．音声認識はニューラルネットワークの発展と大きなコーパスが作られるようになったことにより，単一の End-to-End モデルの学習が可能になった．この End-to-End 音声認識モデルは CNN や GRU を用いた巨大なネットワークとなっている．この巨大ネットワークを様々な用途で用いるために，モデル圧縮する必要があり，3 章と 4 章にて提案する．

第 3 章　Sequence Knowledge Distillation

本章では, 3.1 節において, Knowledge Distillation について述べる．3.2 節において，モデル設計について述べる．3.3 節において，提案する Knowledge Distillation について述べる．

3.1　Knowledge Distillation

巨大なモデル，及び複数のモデルの平均を推論として利用するアンサンブルは必要な計算リソースが多い問題がある．Hinton らは Knowledge Distillation と呼ばれる手法を提案し, 巨大なモデルの情報を小さなモデルに圧縮できることを示した [23]. ここで，ニューラルネットワークの最終層が温度付きソフトマックスである場合を考える．

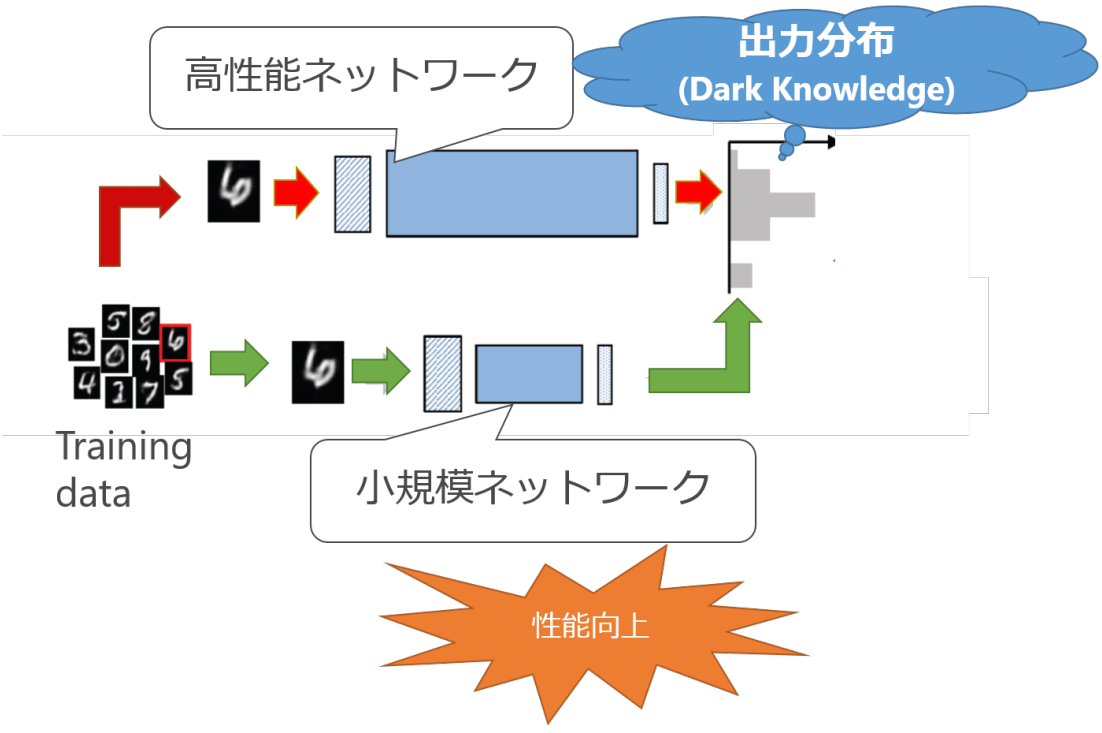


図 3.1　Knowledge Distillation

$$\sigma(\mathbf{z};T)_i = \frac{\exp(z_i/T)}{\sum_{k=1}^{K}\exp(z_k/T)} \tag{3.1.1}$$

ただし，z_i は最後の層への入力，T は温度であり，T=1 の時，通常のソフトマックスとなる．温度が高くなるに従って，分布は平滑化され一様分布に近づく．

あるニューラルネットワークを学習した場合を考える．このモデルは正解ラベルに高い確率を割り当てるが, 正解でないラベルについても 0 ではない確率を割り当てている．例えば，"6"が正解ラベルの時 q_6 が 0.6，q_8 が 0.3，q_6 が 0.01 である場合，"6"と"8"は似ていて，"6"と"1"は似ていないと解釈出来る．このようなラベルの誤り方の情報を Hinton らは 暗黒知識（dark knowledge）と呼んでいる．このような学習済みモデルを教師モデルとし，新たに学習するモデルを生徒モデルとする．また，正解ラベルを示した 1-hot ベクトルをハードターゲット，教師モデルの出力分布をソフトターゲットとする．教師モデルの出力を $f(x)$，生徒モデルの出力を $g(x)$，損失関数を l とした時，以下のような最適化を行う．

$$\text{minimize}_g E_{x\sim p}[l(f(x), g(x))] \tag{3.1.2}$$

式が示す通り，図 3.1 のような教師モデルのソフトターゲットを生徒モデルが学習することは Knowledge Distillation と呼ばれている．この蒸留手法はハードターゲットを用いるより早く学習が終わる [23]，複数のアンサンブルモデルを同等の単一のモデルで表現できるようになる [23, 24]，幅広で浅い学習モデルを教師モデルとして利用し，幅狭で深い生徒モデルを学習させ，教師より高い性能を得る [41] などが知られている．この時，温度パラメータを上げることにより，学習の調整を行うことが出来る．

本研究では，この蒸留アルゴリズムを用いて，モデル圧縮を行う．

3.2 モデル設計

蒸留に用いられる教師ネットワークは，Deep Speech 2[9] で書かれた論文に基づいて構築している．

3.2.1 Deep Speech2

Deep Speech2 は，End-to-End モデル音声認識で，英語と中国語の 2 つの異なる言語を認識するものである．Deep Speech1 の単方向の多層モデルでは，数千時間の音声を学習することができなかった．そこで，大きいデータセットそ学習するために，モデルの容量を大きくする試みをし，CNN，Bidirectinal GRU を使い，モデル化した．これにより，Deep Speech1 のモデルより 8 倍の計算時間がかかるようになった．そこで，いくつかの手法を用いて，学習の収束が早くなる工夫を行った．まず，RNN の入力間で長いストライド (窓関数を t 動かす) を行うことにより，入力を減らした．また，RNN 以外の層で正則化テクニックである Batch Normalization[42] を用い，RNN では全ての時間出力を Batch Normalization する Sequence-wise Batch Normalization[9] を用いることにより，内部共変量シフトの発生を阻止し，学習を加速させた．また，1epoch 目では発話長が短いものから学習させる SortaGrad を行うことにより，1epoch 目の学習が上手くいく変更を行った．これらの工夫により，学習は高速化されたが，依然として，パラメータ数は非常に多く，パラメータを減らす工夫が必要である．

3.2.2 ベースライン設計と提案手法設計

図 3.2 に示す 2 種類の教師ネットワークと 2 種類の生徒ネットワークが用意されている．4 つのモデルすべてで，最初のレイヤーは 1 次元の CNN を使用する．（a）は教師ネットワークの双方向 GRU 層，（b）は GRU 層はでそれぞれユニット数 1510 に設定，（c）の全結合層のユニット数は 1024 であり，1D-CNN も 1024 の出力に揃えている．これらのすべての活性化関数は ReLU を使用している．最終層は全結合層で，Connectionist Temporal Classification(CTC) 損失関数によって最適化されている．

3.3 提案手法

3.3.1 Sequence Knowledge Distillation

本研究では，入力長と出力長と正解ラベル長が異なる，CTC を使った系列学習の蒸留を行う．素直に考えると，出力分布をソフトターゲットとし，各時間ごとに学習する方法を取ることとなる．ところで，Sequence-to-Sequence を蒸留した先行研究で Kim らは，系列ラベルは多くの誤り系列を含むため，温度パラメータではなく，教師モデルの出力分布に正解ラベルの 1-hot ラベルを線形加算することにより調整を行った [25]．本研究でも，CTC の出力を単語の生起確率を表現する木に変換して線形加算することで，保管することは可能である．しかし，これを行うと，蒸留した出力分布を大きく変化させてしまう結果となってしまい，同様の手段をとることができない．そこで，Sequence Knowledge Distillation（SKD）では，学習された教師ネットワークの出力から得られたテキストおよび正しい書き起こし文が，学習のターゲットとして使用される．図 3.3 に示すように，各バッチについて $P\,(\alpha)$ を持つ教師ネットワークの出力を学習し，$P(1-\alpha)$ で正しい文章を学習する．本研究では，$\alpha = 0.5$ を指定した．

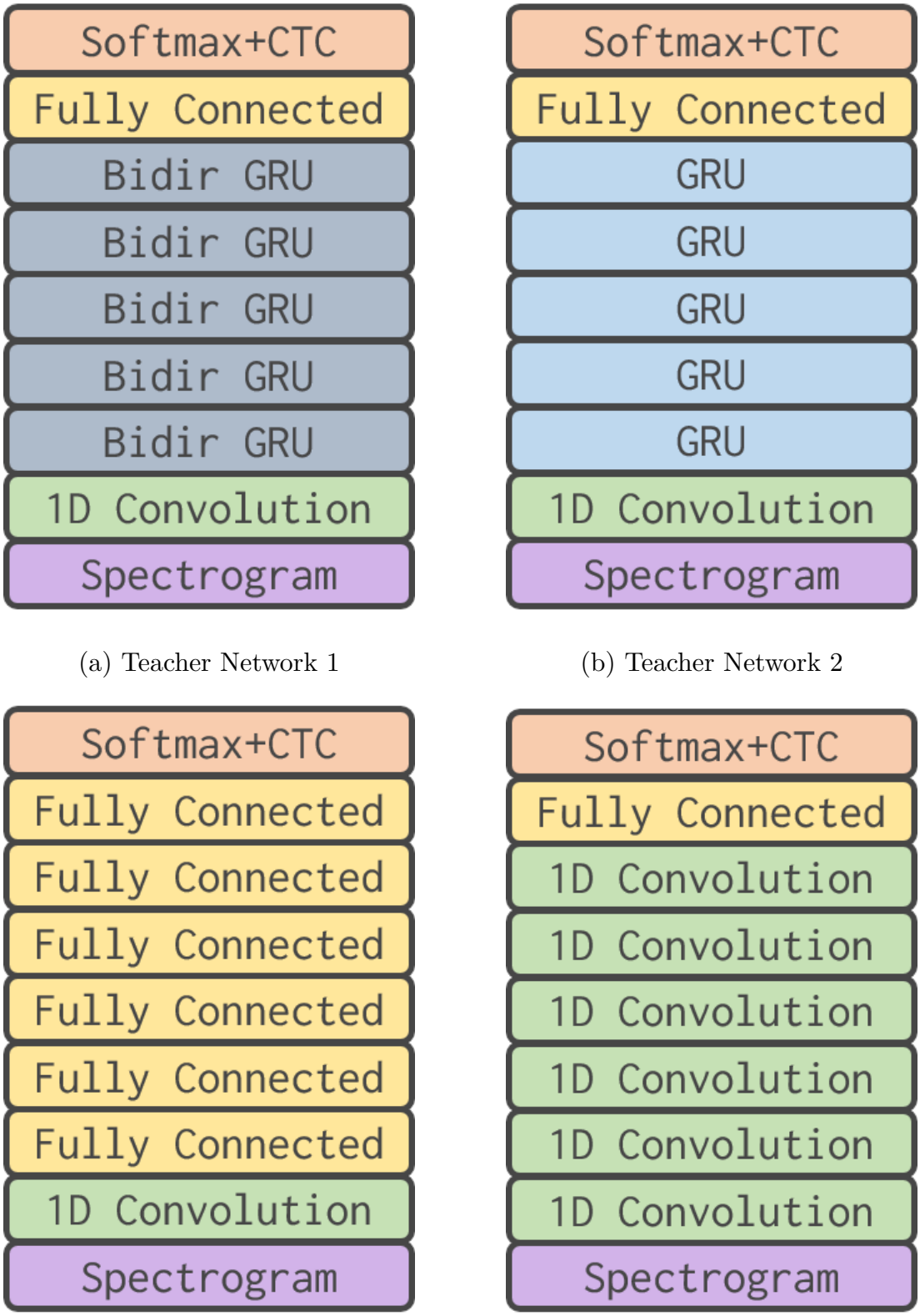
3.3.2 Sequence Hard Ensemble Knowledge Distillation

Sequence Ensemble Hard Knowledge Distillation は，図 3.2 の（a）と（b）の両方の教師ネットワークを使用している．各バッチの $P(\alpha)$ を持つ教師ネットワーク（a）の出力を図 3.4 に示すように学習し，教師ネットワークの出力を $P(\alpha)$，正しい文章を書く．本研究では，$\alpha = 0.25$ を指定した．

3.3.3 Sequence Soft Ensemble Knowledge Distillation

Sequence Ensemble Soft Knowledge Distillation では，図 3.2（a）と図 3.5（b）の両方の教師ネットワークも使用している．図 3.5 に示すように，教師ネットワーク（a）および（b）のソフトマックス関数による出力は，各バッチについて $P(\alpha)$ を算術平均し，テキストに変換して学習し，$P(1-2\alpha)$ に計算される．この研究で

は，$\alpha = 0.5$ とした．



(a) Teacher Network 1

(b) Teacher Network 2

(c) Student Network 1

(d) Student Network 2

図 3.2　Model Architectures

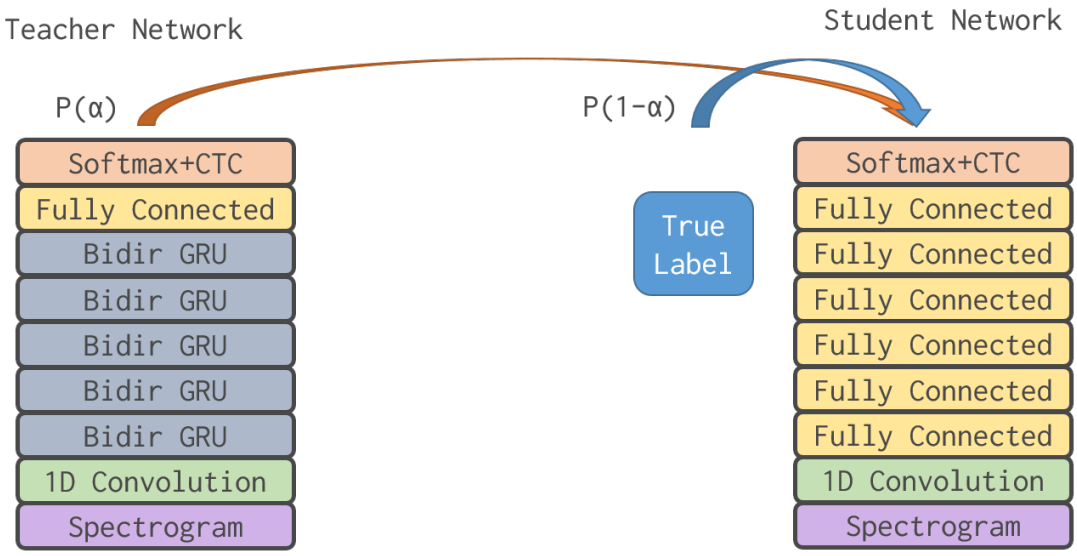


図 3.3　Sequence Knowledge Distillation

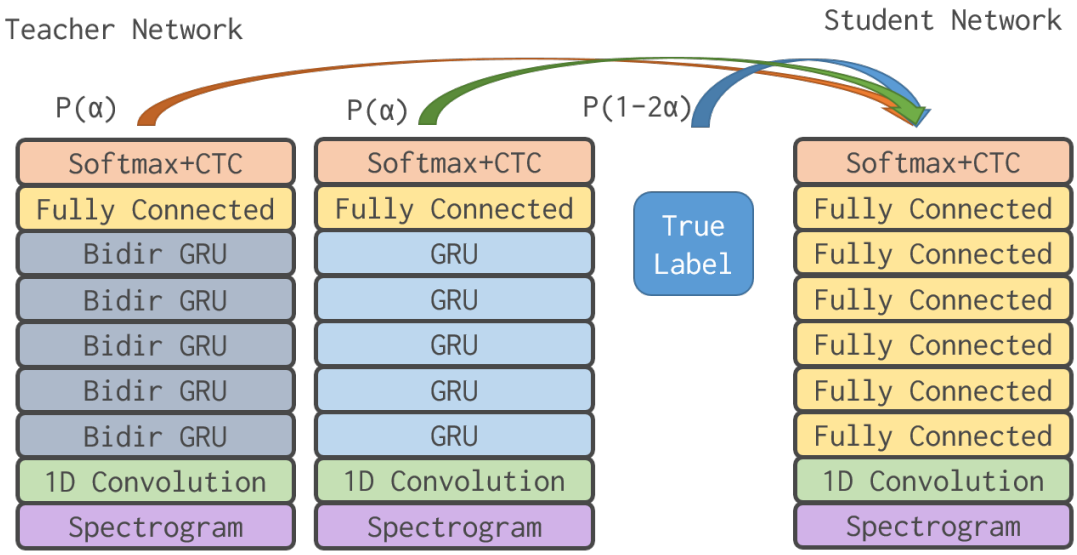


図 3.4　Sequence Hard Ensemble Knowledge Distillation

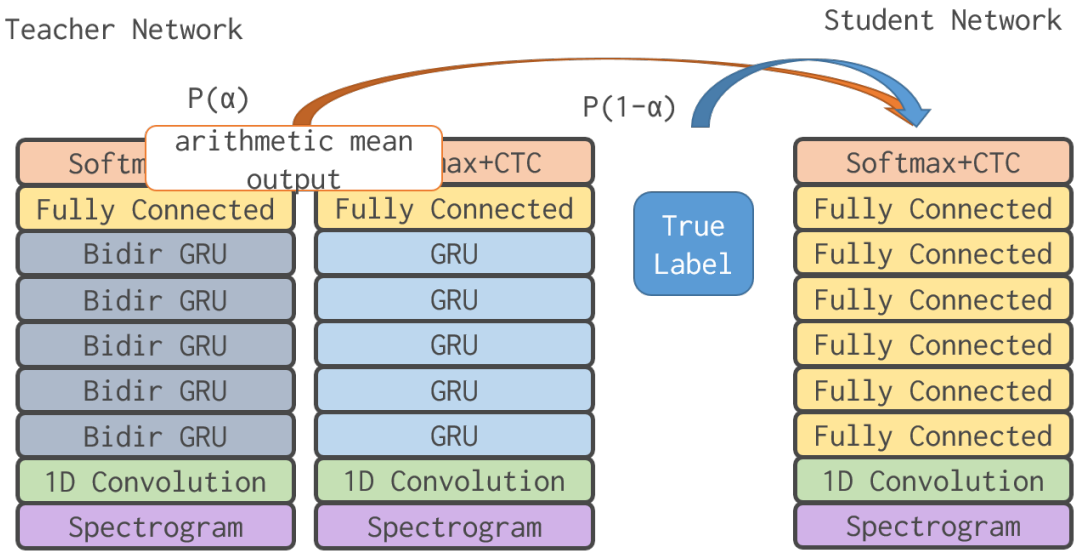


図 3.5　Sequence Soft Ensemble Knowledge Distillation

第 4 章　テンソルトレイン分解

本章では，4.1 節において，テンソル分解について述べる．4.2 節において，モデル設計について述べる．4.3 節において，提案手法について述べる．

4.1　テンソル分解

テンソル分解は多次元配列やテンソルとして表されるデータを処理するための不可欠な手法である．推薦システムや協調フィルタリングで用いられる [43]．テンソル分解を用いることにより，記憶コストを減らし，欠損要素を予測することが知られている [44]．テンソル分解は Tucker 分解 [45] または CP 分解 [46] が一般的に使用される．テンソル分解は，関係抽出によく用いられている [47]．Schein らは国同士の関係性を抽出のために Tucker 分解を用い [48]，Kang らは言語データを主語，動詞，目的語のテンソル化し CP 分解で効率的な分散処理を提案した [49]．また，脳情報データの解析手法に CP 分解が用いられる [50]．近年，マルコフランダム場の推論の近似 [51]，教師あり学習モデル化 [52]，制限ボルツマンマシンの解析 [53]，ニューラルネットワークの圧縮 [19] ために，テンソルトレイン分解 [54] が研究されている．テンソルトレイン分解の特徴はより高階のテンソルの場合，テンソルトレイン分解の特徴は，表現力を保持しながら，テンソルトレイン形式と呼ばれるより省スペースの表現を提供することである．以下に CP 分解と Tucker 分解とテンソルトレイン分解を定式化する．

CP 分解

$$W = \sum_{r=1}^{R} a_r \otimes b_r \otimes c_r \tag{4.1.1}$$

Tucker 分解

$$W = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1 r_2 r_3} \otimes a_{r_1} \otimes b_{r_2} \otimes c_{r_3} \tag{4.1.2}$$

テンソルトレイン分解

$$\mathcal{W}(j_1, j_2, .., j_d) = G_1[j_1] \cdot G_2[j_2] \cdot .. \cdot G_d[j_d]. \tag{4.1.3}$$

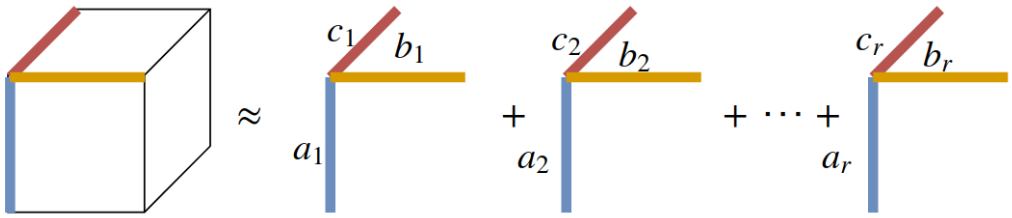


図 4.1　CP 分解

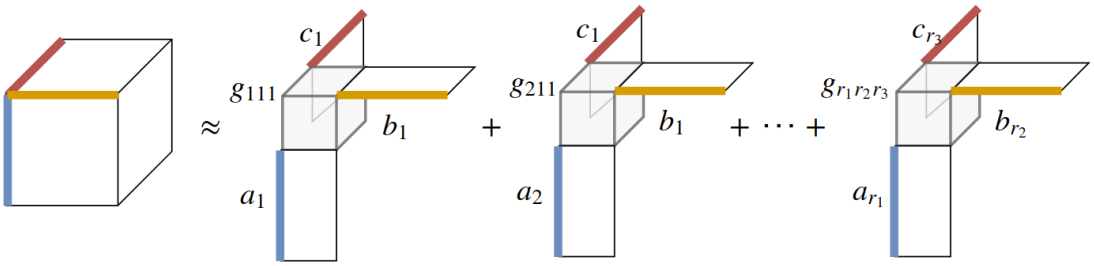


図 4.2　Tucker 分解

K 階テンソルが与えられた場合，Tucker 分解の空間オーダは K で指数関数的であるのに対し，TT 分解の空間オーダは K では線形である．さらに，テンソルトレイン形式では，代数演算を効率的に行うことができる [54]．これらより，テンソルトレイン分解が，高階テンソルを分解する時に非常に効率的であることが分かる．

4.2　モデル設計

CP 分解やタッカー分解は 2 階以下のテンソルを分解する場合は十分高速であるので，深層学習に取り込まれている [55, 56]．重み行列は初期の重みからそれほど動くことなく，無駄な情報を多く持つ [10] と言われているが，重み行列は Lecun の初期化 [57]，Glolot の初期化 [58]，He の初期化 [59]，代表されるように，大雑把に正規分布からサンプリングされたものを使われており，単純に行列の上位ランクを使うことが最適とは言えない．そこで，本研究では，行列を一度高階テンソルに

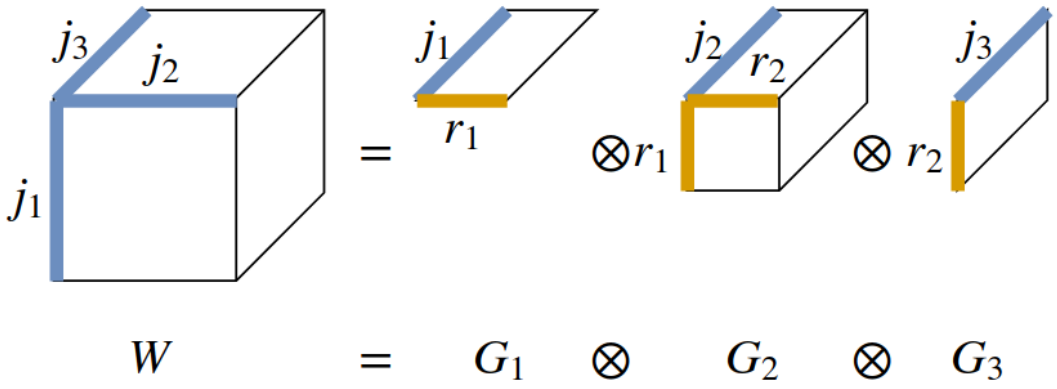


図 4.3　Gated Recurrent Unit の構造

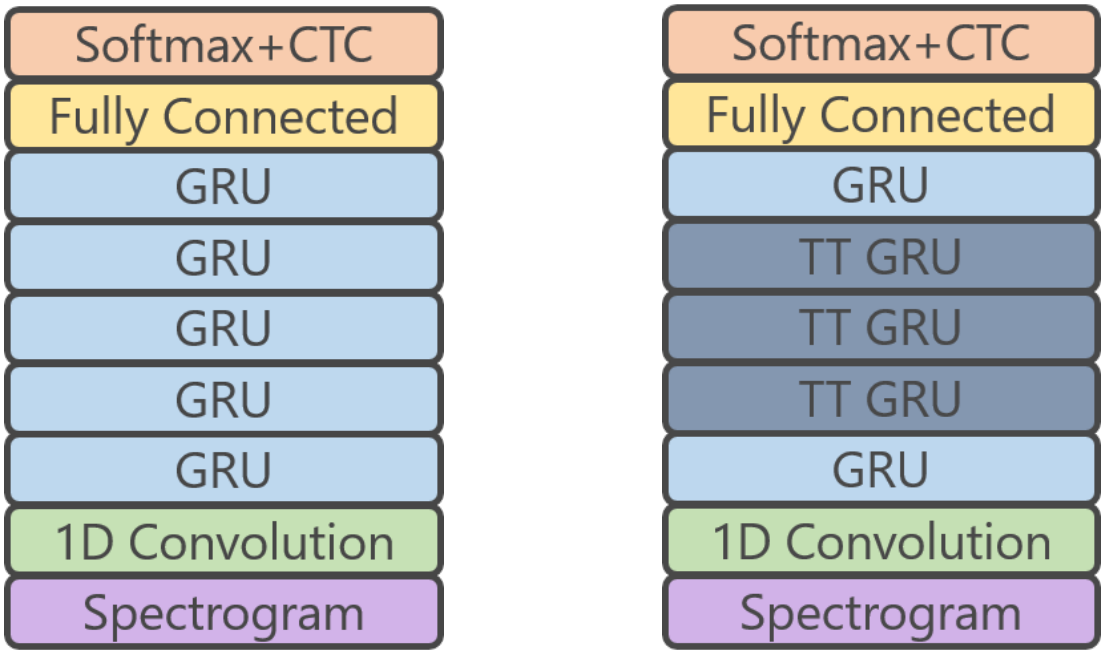
変換し，テンソル分解を行うことで，複雑な表現を可能にし，また，高速化と最適化を行う．ここで，CP 分解やタッカー分解を使う場合，これらのアルゴリズムは，空間オーダ，時間オーダともに，テンソルの階数に対して指数オーダであり，現実的な時間で用いることが出来ないため，テンソルの階数に対して線形オーダであるテンソルトレイン分解を用いる．

本研究は Deep Speech2[9] をベースラインとし，このモデルで用いられた GRU 層を TT GRU 層に置き換えたものである．単純に GRU を TT GRU に置き換える場合，実験的に学習が進まないことが分かったので，中間の 3 層のみを置き換える．具体的な GRU の TT GRU への変換方法を次節で説明する．

4.3　提案手法

4.3.1　行列のテンソル化

行列 $\mathcal{W}$ のテンソル化を行う時，要素数を維持する形で行う．図 4.4 では，$G(2^3, 2^3)$ の形で表される 2 階テンソル (ここでは正方形) が $G(2^2, 2^2, 2^2)$ の形の 3 階テンソル (ここでは立方体) となる．



（a）GRU モデル　　（b）TTGRU モデル

図 4.4　モデル設計

4.3.2　テンソルトレイン分解による GRU の圧縮

本章では，テンソルトレイン分解による GRU の重み行列の圧縮 [20] について解説する．$k \in \{1, .., d\}$ と，k 次元の次元インデックス $j_k \in \{1, .., n_k\}$ の値を用いて表せる，d 次元配列（テンソル）を $\mathcal{W}$ とする．$\mathcal{W}$ の要素が以下の式で行列 $G_k[j_k]$ によって表せるときテンソルトレインフォーマットと呼び，以下の式で定式化される [54].

$$\mathcal{W}(j_1, j_2, .., j_d) = G_1[j_1] \cdot G_2[j_2] \cdot .. \cdot G_d[j_d]. \tag{4.3.1}$$

例えば，$d = 2$ のときのテンソルフォーマット適用を図示すると，図 4.6 のように表すことができ，低ランク近似するときに必要となる $r_k \in \{1, .., n_k\}$ がテンソルトレインランクである．

GRU では，入力から隠れ層，隠れ層から出力層が 1 つずつ，リセットゲートを制御する重みが入力と出力で 2 つ，更新ゲートを制御する重みが入力と出力で 2 つ

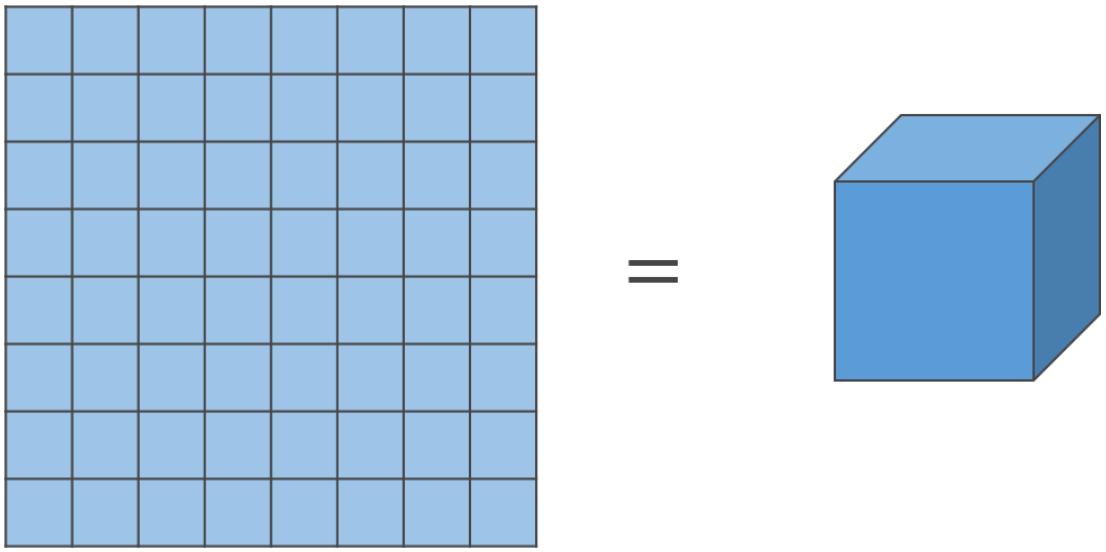


図 4.5　行列のテンソル化

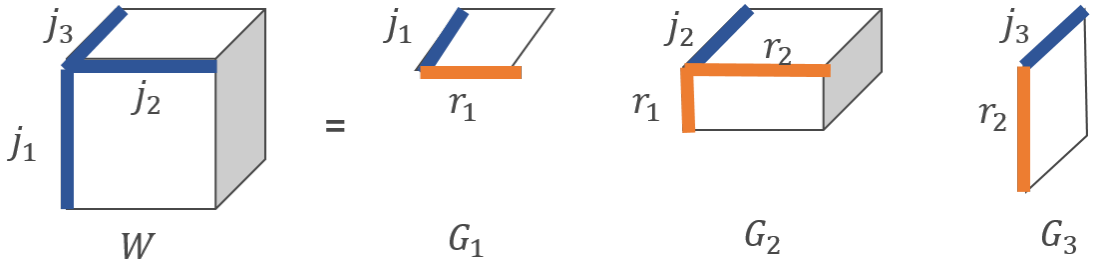


図 4.6　テンソルトレインフォーマットによる低階テンソル表現

の合計 6 つの重み行列で表すことができる．以下に，入力から隠れ層，隠れ層から出力の重みの圧縮について説明するが，他の 4 つの重みも同様の手法で圧縮することができる．

入力から隠れ層の重みを行列 $W_{xh} \in \mathbb{R}^{M \times M}$ と定義し，隠れ層から出力の重みを行列 $W_{hh} \in \mathbb{R}^{M \times M}$ と定義する．今回，GRU に対する入力次元と出力次元が同じであるため，どちらの重みも $\mathbb{R}^{M \times M}$ に帰属する．本手法では，この (W_{xh}, W_{hh}) を圧縮する．

まず，行列の形状 M を $\prod_{k=1}^{d} m_k$ に分解する． 次に，モデルのテンソルトレインランク $\{r_k\}_{k=0}^{d}$ を決定し，W_{xh} をテンソル $\mathcal{W}_{xh}$ と W_{hh} をテンソル $\mathcal{W}_{hh}$ と置き換

える．テンソル $\mathcal{W}_{xh}$ は，$\forall k \in \{1,..,d\}$，$\mathcal{G}_k^{xh} \in \mathbb{R}^{m_k \times m_k \times r_{k-1} \times r_k}$ である時，テンソルトレインコア $\{\mathcal{G}_k^{xh}\}_{k=1}^d$ で表すことができ，テンソル $\mathcal{W}_{hh}$ は，$\forall k \in \{1,..,d\}$，$\mathcal{G}_k^{hh} \in \mathbb{R}^{m_k \times m_k \times r_{k-1} \times r_k}$ である時，テンソルトレインコア $\{\mathcal{G}_k^{hh}\}_{k=1}^d$ で表すことができる（図 1）．また，W_{xh} および W_{hh} のテンソルコアによって定義される行 $p \in \{1,..,M\}$ を全単射する関数 $\mathbf{f}_i^x$ および $\mathbf{f}_i^h$ を定義する．これらを用いて，RNN の出力 h_t は以下のように書き表せる．

$$a_t^{xh}(p) = \sum_{j_1,..,j_d} \mathcal{W}_{xh}(\mathbf{f}_i^x(p), [j_1,..,j_d]) \cdot \mathcal{X}_t\,(j_1,..,j_d) \tag{4.3.2}$$

$$a_t^{hh}(p) = \sum_{j_1,..,j_d} \mathcal{W}_{hh}(\mathbf{f}_i^h(p), [j_1,..,j_d]) \cdot \mathcal{H}_{t-1}\,(j_1,..,j_d) \tag{4.3.3}$$

$$a_t^{xh} = \left[a_t^{xh}(1),..,a_t^{xh}(M)\right] \tag{4.3.4}$$

$$a_t^{hh} = \left[a_t^{hh}(1),..,a_t^{hh}(M)\right] \tag{4.3.5}$$

$$h_t = f(a_t^{xh} + a_t^{hh} + b_h), \tag{4.3.6}$$

$\mathcal{X}$ は入力 x_t のテンソル表現であり，$\mathcal{H}_{t-1}$ は以前の隠れ状態 h_{t-1} のテンソル表現である．以上が GRU の入力から隠れ層，隠れ層から出力の重みの圧縮であり，GRU で圧縮するには，リセットゲートと更新ゲート内の行列を同様の方法で圧縮することで達成できる．これをテンソルトレイン GRU（TTGRU）とする．

第 5 章　実験

本章では，5.1 節において，実験データについて述べる．5.2 節において，特徴抽出について述べる．5.3 節において，学習規則について述べる．5.4 節において，実験結果を述べる．

5.1　実験データ

表 5.1　Libri Speech の概要

subset	hours	speakers	per-spk minutes
train-clean-360	363.6	921	25
train-clean-100	100.6	251	25
dev-clean	5h	40	10
test-clean	5h	40	10

我々は，このモデルを評価するタスクとして，英語音声コーパスである LibriSpeech コーパス [10]（参照）を使用した．この実験では，LibriSpeech コーパスのいくつかのサブセットが使用され，使用されるコーパスは表 5.1 に示されている．トレーニングデータに train-clean-1000 サブセット，検証データに dev-clean サブセット，テストデータにの test-clean サブセットを用いた．

5.2　特徴抽出

音声から音声認識に必要な特徴量を抽出する処理である. 特徴量抽出の役割は計算資源の節約とノイズ除去である．一般に,AD 変換された波形を入力とし, 一定の時間間隔で, 特徴量を抽出する. その際, 一定間隔ごとに分析される対象を音声フレームと呼ぶ．従来の手法では，メル周波数ケプストラム係数（Mel-Frequency Cepstrum Coefficients, MFCC）[60] という対数パワースペクトルに対しフィルタをかけてフイルタバンク群を構成し, その出力に対し離散コサイン変換を行い, その

結果から低次の項, すなわち, スペクトル包絡の成分を取り出したものがよく用いられていた. しかし，CNN は波形に対する畳み込み演算を行うことで, フーリエ変換に代わる周波数解析を行っているといわれており，本稿では，パワースペクトルで正規化したスペクトログラムを用いている [9].

5.3 学習規則

本節では本研究で用いた損失関数 CTC と学習方法である SortaGrad の説明を行う.

5.3.1 コネクショニスト時系列分類法

入出力間で系列長が違う場合の分類問題を, 上述のように HMM を使うことなくニューラルネットだけで解決しようとするのが, コネクショニスト時系列分類法 (Connectionist Temporal Classification，CTC) である [61, 62, 63]．CTC は, RNN の出力の解釈を変更し, 入力系列と長さの違う出力系列を扱えるようにし，任意の RNN に適用可能である．認識対象となるラベルの集合を l とする．これに空白を表す ϕ を追加した冗長ラベル集合 l' を作成．入力時間が T=8 の時，図 5.1 のようなパスが作成される．この時，図の赤線で書かれたパスのラベル $\pi = \{\phi c \phi a a \phi \phi t\}$ である．入力データ列を $x = x^1, x^2, ..., x^T$，出力を $y = y^1, y^2, ..., y^T$，Softmax への入力を $u = u^1, u^2, ..., u^T$ の時，次のように定式化される.

ラベル列の確率

$$p(\boldsymbol{\pi} \mid \mathbf{x}) = \prod_{t=1}^{T} y_{\pi_t}^t \tag{5.3.1}$$

ラベル l がとりうるパスの総和

$$p(\boldsymbol{l} \mid \mathbf{x}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{l})} p(\boldsymbol{\pi} \mid \mathbf{x}) \tag{5.3.2}$$

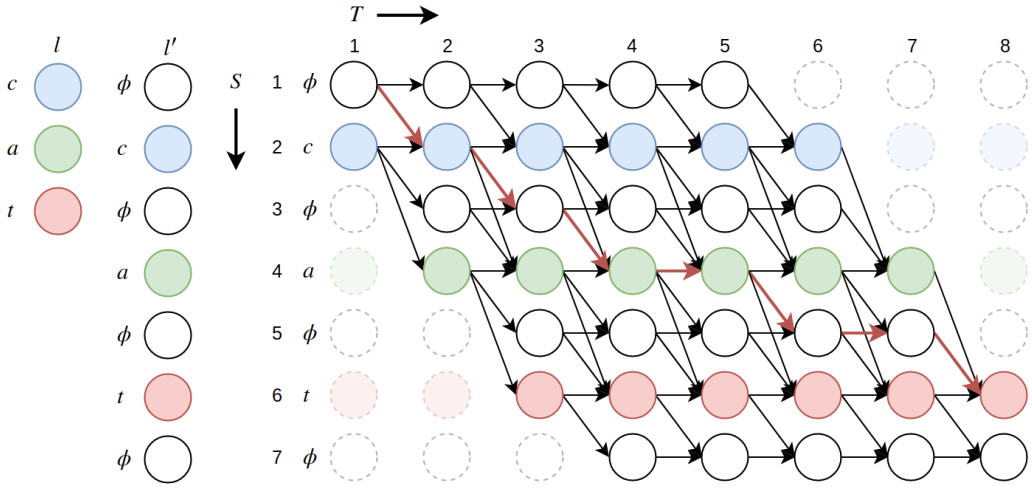


図 5.1　Connectionist Temporal Classification

前向き確率

$$\alpha_t(s) \stackrel{\text{def}}{=} \sum_{\mathcal{B}(\boldsymbol{\pi}_{\infty:\sqcup})=\boldsymbol{l}_{\infty:\lfloor s/2 \rfloor}} \prod_{t'=1}^{t} y^{t'}_{\boldsymbol{\pi}_{t'}} \tag{5.3.3}$$

後ろ向き確率

$$\beta_t(s) \stackrel{\text{def}}{=} \sum_{\mathcal{B}(\boldsymbol{\pi}_{\sqcup:T})=\boldsymbol{l}_{\lfloor s/2 \rfloor:|\boldsymbol{l}|}} \prod_{t'=t}^{T} y^{t'}_{\boldsymbol{\pi}_{t'}} \tag{5.3.4}$$

パスの確率

$$\alpha_t(s)\beta_t(s) = \sum_{\boldsymbol{\pi}\in\mathcal{B}^{-1}(\boldsymbol{l}),\pi_t=l'_s} y^t_{l'_s} \prod_{t=1}^{T} y^t_{\pi_t} \tag{5.3.5}$$

5.3.2　SortaGrad

音声認識や機械翻訳を含む多くの系列学習タスクの共通のテーマとして，長い系列の学習が困難になる傾向がある [64]．また，CTC を用いた音声認識では，長い発話は損失関数の値が大きくなるため，困難なタスクであるといえる．一方で，難易

度順に例を提示することでオンライン学習が加速されることを発見されており，カリキュラムラーニング [65, 66] に基づいたものである．そこで，長い発話を難易度の高いタスクとし，1 エポック目の学習では発話長が昇順になる学習，SortaGrad[9] を学習規則として採用した．

5.4 Libri Speech による評価

5.4.1 Sequence Knowledge Distillation

この章では，Sequence Knowledge Distillation(SKD)，Sequence Soft Ensemble Knowledge Distillation(SSEKD)，Sequence Hard Ensemble Knowledge Distillation(SHEKD) で訓練した DNN と CNN モデルを評価し，通常の学習をしたベースライン（簡単な DNN と CNN）を比較する．この実験では，LibriSpeech コーパスの一部のサブセットを使用する．使用されるコーパスは ref tab：dataset に示した．学習データに train-clean-360 サブセット，検証データに dev-clean サブセット，SKD を評価するための評価データに test-clean サブセットを使用した．さらに，学習データに train-clean-100 サブセット，検証データ用に dev-clean サブセット，SHEKD と SSEKD を評価用データとして使用した．SKD のタスクでは，図 5.2（a）は教師モデルとして用いられ，SHEKD と SSEKD のモデルでは，図 5.2（a）と図 5.2（b）の生徒モデルを使用した．蒸留された生徒モデルのために，図 2 の FC ベースのモデルを使用する． FC のモデルでは，時間方向の順序が小さく，速く，パラメータが小さい．CNN のモデルは，パラメータが増加しているものの，時間方向に秩序が低下するため，高速であるという特徴がある．学習中に検証データの損失が最も減少した時点でモデルを評価した．この音声認識タスクでは，16kHz でサンプリングされた発話が入力として与えられ，正しいラベルである文字の誤り率によって評価した．

実験の結果，SKD により Table5.4 と Table5.3 の殆どのタスクで精度が向上した．これは，教師モデルのモデル出力が統計に基づくノイズとして機能しているためである．また，雑音のパターンを増加させた SHEKD は，SKD とほぼ同等またはそれ以上の精度で改善することに成功した．SSEKD は，精度が向上したものとそうでないものに分かれ，いずれも SKD の上昇幅に達していない．これは CTC

表 5.2　Validation with 360-hour dataset

train-clean-360	Model	Param	Compr. Ratio	CER
Teacher	BGRU	24M	100%	9.60%
Baseline	CNN	59M	246%	24.75%
	FC	7M	29.20%	45.10%
SKD	CNN	59M	246%	25.06%
	FC	7M	29.20%	43.58%

表 5.3　Validation with 100-hour dataset

train-clean-100	Model	Param	Compr. Ratio	CER
Teacher	BGRU	24M	100%	23.43%
	GRU	24M	100%	22.58%
Baseline	CNN	59M	246%	35.05%
	FC	7M	29.20%	50.24%
SKD	CNN	59M	246%	34.76%
	FC	7M	29.20%	49.47%
SHEKD	CNN	59M	246%	33.49%
	FC	7M	29.20%	49.88%
SSEKD	CNN	59M	246%	34.48%
	FC	7M	29.20%	51.81%

の特性によるもので，例えば，最終層からのデコードが $\phi\phi\phi\phi CAT$ になっている場合と，$CAT\phi\phi\phi\phi$ となっている場合，どちらも CAT という出力を得るが，これを加算平均した場合，$CAT\phi\phi CAT$ となってしまい，デコードが $CATCAT$ となってしまう場合があるからである．

5.4.2 Tensor Train Decomposition

この音声認識タスクでは，16kHz でサンプリングされた発話が入力として与えられ，正しいラベルである文字列誤り率で評価される．TTGRU のモデルパラメータを最適化するために Adam アルゴリズム [67]，ベースラインの GRU には SGD アルゴリズム（最急降下法）を使用した．また，ベースラインモデルでは，収束高速化のための Layer-wised Batch Normalization を用いている [9].（最適化手法の違いが実験結果に与える影響がわかるなら述べてください）以降，GRU は GRU-HF と表し，TTGRU は TTGRU-HF-R で表す．ここで，F は隠れユニットの数であり．R はテンソルトレインランクを表す．H は隠れ層を表す．例えば，TTGRU-H4x8x6x8-R3 は，テンソルトレインフォーマットのテンソルトレインランク 3 で 4x8x6x8 の隠れユニットを有する TTGRU を意味する．GRU のパラメータの設定は Deep Speech2[9] の論文に基づいており，TTGRU のパラメータの設定は検証セットのパフォーマンスに基づいて決定され，図 4.4 のような設計となっている．

表 5.4　train-clean-100 による学習の結果

Model	Param	Comp	Val CER	Test CER
GRU-H1510	13M	100	20.03%	20.62%
TTGRU H4x8x6x8-R3	11k	0.08	27.57%	27.21%
TTGRU H4x8x6x8-R5	22k	0.16	23.76%	23.40%
TTGRU H4x8x6x8-R7	37k	0.27	22.68%	23.73%

実験の結果と GRU のパラメータがどの程度圧縮されているかを表 5.4 に示した．TTGRU はランク 5 やランク 7 では圧縮される前のモデルに近い性能を引き出すことに成功した．ベースラインモデルは，1510 の隠れユニットを持つ GRU である．我々の提案するモデルは，4x8x6x8 の出力形状とテンソルトレインランク $(3, 5, 7)$ の TTGRU である．音声入力はスペクトログラムに変換して正規化したもたのである．評価関数には単語誤り率 (Character Error Rate, CER) を用いた．表 2 は, ベースラインおよび提案されたモデルに関する我々の実験の結果である．表は，これらのモデルが検証セットとテストセットが同様の性能を有することを示

している．我々の提案したモデルは，変換した GRU 層に限れば約 99% 削減し，モデル全体では約 60% 削減しているが，パラメータの数を減らすと同時に性能を維持することができた．

第 6 章　総括

6.1　本論文のまとめ

本稿では，CTC を用いた，高性能な End-to-End の音声認識の蒸留を行った．一般的に End-to-End 音声認識では大規模な深層学習を必要とするが，本研究で提案した方法では小さなモデルの精度を向上させると同時に，確率的にラベルを切り替えることにより過学習が防止される．

また，本稿では，CTC ベースの End-To-End 音声認識においてテンソルトレイン分解が有効であることを示した．テンソルトレインフォーマットを使用して，複数の低ランクテンソルをによって GRU レイヤー内の重み行列を表現した．TTGRU End-to-End 自動音声認識は，モデルの性能と精度を維持しながらパラメータの数を大幅に圧縮した．我々は，読み上げコーパスでこれを評価し，難しいタスクでも精度を失うことなく元のモデルと比較して RNN パラメータを大幅に削減できた．

6.2　今後の展望

本稿で提案した Sequence Knowledge Distillation は，データが多くなると，蒸留による効果が薄くなるという問題がある．そこで，出力分布を蒸留する，強い制約と出力文を学習する弱い制約を用いて最適化することが課題である．また，Sequence Knowledge Distillation と Tensor Train Distillation は違う手法のモデル圧縮であるため，これらを両立した実験を行いたい．

テンソルトレイン分解を用いたニューラルネットワークは，分解後のパラメータの数や，通常の計算速度は高速であるが，誤差逆伝搬で最適化する際には，他のテンソル分解の CP 分解や Tucker 分解と同様に，高い計算量になってしまう問題を残している．しかし，テンソルトレイン分解の最適化の高速化の研究も行われており [44]，この誤差逆伝搬にも適用可能であり，これを利用した高速化も今後の展望である．また，本稿では，言語モデルを使用していないモデルを用いているが，言語モデルを用いた最適化や，言語モデルを含めた圧縮も今後の展望である．

謝辞

本学情報科学研究科の中村哲教授には，主指導教官として数多くの貴重なご助言を頂いたほか，充実した研究環境の整備など，研究活動を手厚くご支援いただき，心より感謝を申し上げます．

本学情報科学研究科の松本裕治教授には，副指導教官として様々なご助言を頂きました．心より感謝を申し上げます．

本学情報科学研究科の Sakriani Sakti 特任准教授には，副指導教官として，研究の方向性など数多くのご助言を頂いた他，コーパスの調達や研究会発表の機会をいただき，心より感謝を申し上げます．

本学情報科学研究科の吉野幸一郎助教には，数多くの貴重なご助言を頂いたほか，学会参加時にはサポートしていただきました．心より感謝を申し上げます．

本学情報科学研究科の須藤克仁准教授，田中宏季助教には，数多くの貴重なご助言を頂きました．心より感謝を申し上げます．

本学情報科学研究科の Andros Tjandra 様，叶高朋様をはじめ，SP 班の方々には数多くの貴重なご助言を頂きました．心より感謝を申し上げます．

また，本学知能コミュニケーション研究室秘書の松田真奈美様，林美保様には諸手続きを始めとする様々な面でお世話になりました．心より感謝を申し上げます．

知能コミュニケーション研究室一同には日頃の議論を通じて研究に関する多くの知識や示唆を頂きました．心より感謝を申し上げます．

最後に，学業生活を様々な面から支えてくれた家族に感謝致します．

参考文献

[1] B.-H. Juang and L. R. Rabiner, "Automatic speech recognition–a brief history of the technology development," *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*, vol. 1, p. 67, 2005.

[2] 河原達也, "知識の森 2 群-7 編-2 章 音声認識," **電子情報通信学会**, vol. 7.

[3] D. R. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[4] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proc. sigir*, vol. 7, 2007, pp. 51–57.

[5] 篠田浩一, **音声認識** *(***機械学習プロフェッショナルシリーズ***)*. 講談社, 12 2017.

[6] 荒木雅弘, **イラストで学ぶ 音声認識** *(KS* **情報科学専門書***)*. 講談社, 1 2015.

[7] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

[8] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[9] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning*, 2016, pp. 173–182.

[10] M. Denil, B. Shakibi, L. Dinh, N. de Freitas *et al.*, "Predicting parameters in deep learning," in *Advances in Neural Information Processing Systems*,

2013, pp. 2148–2156.

[11] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*, 2015, pp. 1135–1143.

[12] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[13] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on.* IEEE, 2013, pp. 6655–6659.

[14] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," in *Advances in Neural Information Processing Systems*, 2014, pp. 1269–1277.

[15] Y. Wang, J. Li, and Y. Gong, "Small-footprint high-performance deep neural network-based speech recognition using split-vq," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on.* IEEE, 2015, pp. 4984–4988.

[16] R. Prabhavalkar, O. Alsharif, A. Bruguier, and L. McGraw, "On the compression of recurrent neural networks with an application to lvcsr acoustic modeling for embedded speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on.* IEEE, 2016, pp. 5970–5974.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, "Tensorizing neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 442–450.

[20] A. Tjandra, S. Sakti, and S. Nakamura, "Compressing recurrent neural network with tensor train," *arXiv preprint arXiv:1705.08052*, 2017.

[21] D. Lin, S. Talathi, and S. Annapureddy, "Fixed point quantization of deep convolutional networks," in *International Conference on Machine Learning*, 2016, pp. 2849–2858.

[22] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 535–541.

[23] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[24] S. R. Bulò, L. Porzi, and P. Kontschieder, "Dropout distillation," in *International Conference on Machine Learning*, 2016, pp. 99–107.

[25] Y. Kim and A. M. Rush, "Sequence-Level Knowledge Distillation," 2016. [Online]. Available: http://arxiv.org/abs/1606.07947

[26] X. He, L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition," *IEEE Signal Processing Magazine*, vol. 25, no. 5, 2008.

[27] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer Science & Business Media, 2012, vol. 247.

[28] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[29] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[30] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent nn: first results,"

arXiv preprint arXiv:1412.1602, 2014.

[31] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[32] M. I. Jordan, "Serial order: A parallel distributed processing approach," *Advances in psychology*, vol. 121, pp. 471–495, 1997.

[33] 岡谷貴之, 深層学習 *(*機械学習プロフェッショナルシリーズ*)*. 講談社, 4 2015.

[34] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[35] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," in *Readings in speech recognition*. Elsevier, 1990, pp. 393–404.

[36] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[37] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei, "Imagenet large scale visual recognition competition 2012 (ilsvrc2012)," 2012.

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[39] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[40] D. Palaz, R. Collobert *et al.*, "Analysis of cnn-based speech recognition system using raw speech as input," Idiap, Tech. Rep., 2015.

[41] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference*

on Machine Learning, 2015, pp. 448–456.

[43] 須山敦志, **機械学習スタートアップシリーズ ベイズ推論による機械学習入門** *(KS* **情報科学専門書***)*. 講談社, 10 2017.

[44] M. Imaizumi, T. Maehara, and K. Hayashi, "On tensor train rank minimization: Statistical efficiency and scalable algorithm," in *Advances in Neural Information Processing Systems*, 2017, pp. 3933–3942.

[45] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[46] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Studies in Applied Mathematics*, vol. 6, no. 1-4, pp. 164–189, 1927.

[47] 石黒勝彦 and 林浩平, **関係データ学習** *(***機械学習プロフェッショナルシリーズ***)*. 講談社, 12 2016.

[48] A. Schein, J. Paisley, D. M. Blei, and H. Wallach, "Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1045–1054.

[49] U. Kang, E. Papalexakis, A. Harpale, and C. Faloutsos, "Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 316–324.

[50] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[51] A. Novikov, A. Rodomanov, A. Osokin, and D. Vetrov, "Putting mrfs on a tensor train," in *International Conference on Machine Learning*, 2014, pp. 811–819.

[52] A. Novikov, M. Trofimov, and I. Oseledets, "Exponential machines," *arXiv preprint arXiv:1605.03795*, 2016.

[53] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, "On the equivalence of restricted boltzmann machines and tensor network states," *arXiv preprint arXiv:1701.04831*, 2017.

[54] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[55] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank rnn language model," *arXiv preprint arXiv:1711.03953*, 2017.

[56] L. Kaiser, A. N. Gomez, and F. Chollet, "Depthwise separable convolutions for neural machine translation," *arXiv preprint arXiv:1706.03059*, 2017.

[57] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 1998, pp. 9–50.

[58] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[60] B. Logan *et al.*, "Mel frequency cepstral coefficients for music modeling." in *ISMIR*, vol. 270, 2000, pp. 1–11.

[61] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[62] A. Graves *et al.*, *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, vol. 385.

[63] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained hand-

writing recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[64] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[65] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[66] W. Zaremba and I. Sutskever, "Learning to execute," *arXiv preprint arXiv:1410.4615*, 2014.

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

発表リスト

国際会議 [査読有]

[1] Keisuke Kinoshita, Marc Delcroix, Haeyong Kwon, Takuma Mori and Tomohiro Nakatani. Neural network-based spectrum estimation for online WPE dereverberation. In Interspeech 2017.

研究会 [査読無]

[1] 森 巧磨, Andros Tjandra, Sakriani Sakti, 中村 哲. テンソルトレイン分解による End-to-End 自動音声認識モデルの圧縮. 研究報告音声言語情報処理（SLP）,2017-SLP-119(18),1-4 (2017-12-14), 2188-8663.