一般社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-057-10 (11/20)

複雑なニューラルネットワークを対象とした ノードプルーニングベースのモデル圧縮の検討

Model Compression of Non-uniform Neural Networks with Non-Linear Functions Based on Node Pruning

中臺 一博 [1,2*]　福本 陽典 [1]　武田 龍 [3]
Kazuhiro NAKADAI[1,2]　Yosuke FUKUMOTO[1]　Ryu TAKEDA[3]

[1] （株）ホンダ・リサーチ・インスティチュート・ジャパン
[1] Honda Research Institute Japan Co., Ltd.
[2] 東京工業大学 工学院 システム制御系　[3] 大阪大学 産業科学研究所
[2] Tokyo Institute of Technology　[3] Osaka University

Abstract: 本稿では，近年複雑化しているニューラルネットワークに対応するため，ノード枝刈りに基づいた不均一で複雑な深層学習ネットワークのモデル圧縮を扱う．ノード枝刈りによるモデル圧縮は，比較的古くから研究されているが，そのほとんどは，活性化関数としてシグモイド関数を用い，バイパス接続のない均一で単純な全結合ニューラルネットワークであることを前提としている．近年は，活性化関数として ReLU など非シグモイド関数を用いることが一般的であるため，こうした非線形活性化関数に対応するため，ノードエントロピー法を拡張したノード活性度推定法を提案する．さらに，バイパス接続など不均一なトポロジを持つネットワークに対応するため，層間ペアリング，バイパス接続の枝刈り，ネットワーク全体に対する枝刈りポリシーに関する規範を提案する．これらを組み合わせた手法を提案手法とし，ReLU，バイパス接続を有する TDNN ベースのニューラルネットワークである，音声認識システム Kaldi 用の音響モデルの圧縮に適用を試みた．結果として，音声認識性能を維持しつつ，音声認識システム全体として 31% の速度向上を達成することができた．

1　はじめに

近年，最適な数以上のパラメーターを使用すれば，深層学習のニューラルネットワークモデルを問題なく学習できることが報告されており，過剰パラメータ化 (over-parameterization) [1, 2, 3] と呼ばれている．一方，過剰パラメータ化によるモデルは規模が大きいため，組み込みデバイス，モバイルコンピューティング，車載情報 (In-Vehicle Information, IVI) システムへの応用など，実用向けには，モデル圧縮によるコンパクト化が不可欠である．また，モデル圧縮により，ネットワークの複雑さや過剰適合問題を軽減することもできる [4]．このため，因子分解 (Factorization), 知識蒸留 (Knowledge Distillation)，枝刈り (Pruning) といったモデル圧縮手法が研究されてきた [5].

因数分解はモデルのスパース性を利用しており，特異値分解 [6]，低ランク行列因子分解 [7]，ベクトル量子化 [8] を利用したモデル圧縮手法が報告されている．また，テプリッツ行列を使用した分解 [9] と，小さい行列の組合せで置換する手法 [10] も，因数分解ベースのモデル圧縮手法と捉えることができる．これらは，画像分類と音声認識 (ASR) を対象に，元のモデル性能を維持または改善しつつ，モデルサイズを大幅に圧縮している．因子分解は，パラメータ行列として表せるネットワークに適用できるので，一般に，比較的単純で均一な全結合型のニューラルネットワークに適用される．複雑なネットワークに適用する場合には，パラメータ行列に変換できる部分を抽出しての適用する必要があるため，限定的な適用となる．

知識蒸留は，主に TS (Teacher-Student) 学習を適用することでモデル圧縮を行う [11, 12]．まず，パラメータ数の多い大規模ニューラルネットワークモデルを，教師モデルとして学習する．次に，パラメータ数が少ない小規模のニューラルネットワークを生徒モデルとして，教師モデルと同等の性能となるように学習を行う．この際の教師モデルと生徒モデルの比較には，KL(Kullback-Liebler) 距離ベースの出力分布 [13] や

*（株）ホンダ・リサーチ・インスティチュート・ジャパン
〒 351-0188 埼玉県和光市本町 8-1
E-mail: nakadai@jp.honda-ri.com

シーケンスレベルの出力分布 [14] などが使用される．知識蒸留では，ラベルを平滑化するような正則化により，効果的なモデル圧縮が実現されているという報告もあるが [15]，教師が与えられた際に，どのようなトポロジーを持った生徒モデルが最適なのかといった問題について十分な議論されているわけではない．

枝刈りは，音声認識 [16, 17, 18]，画像分類 [19, 20, 21, 22, 23]，翻訳 [24] などを対象に広く研究されている．主に，枝刈りの鍵として重みパラメータやノード活性度を用いており，対象となるニューラルネットワークの層，チャネル，ノード，リンクの寄与に応じて，各々が枝刈りもしくは共有化される．性能を維持しつつ，処理速度を向上するために，ビット量子化の併用も試みられている [18, 21, 25]．武田ら [25] は，重みパラメータ，ノード活性度，ビット量子化の組合せにより，ノード枝刈り率 30% で，音声認識精度を維持しつつ，デコード処理の 5 倍高速化を実現している．これらの手法は，枝刈り後に再学習が必要ではあるが，ニューラルネットワークのサイズを大幅に縮小することができる．さらに，因子分解や TS 学習で必要なトポロジーに対する仮定が不要であるため，適用が容易であるという利点がある．しかし，これらの研究では，これまで，DNN や CNN といった単純なニューラルネットワークを対象に，活性化関数には伝統的なシグモイド関数を想定して研究が行われてきた．近年では，バイパス接続など，ネットワークトポロジーは複雑化する傾向にあり，活性化関数にも ReLU [26] など，他の非線形関数が使用されることが一般的になってきている．

つまり，従来のアプローチの問題は，モデルが複雑化しているにもかかわらず，単純で均一なニューラルネットワークを対象としており，不均一かつ複雑なネットワークに対するモデル圧縮のガイドラインが十分に議論されていないことであると言える．そこで，本稿では，バイパス接続や非シグモイド活性化関数を備えた不均一で複雑なネットワークをモデル圧縮する方法を提案する．モデル圧縮のアプローチとしては，因子分解や知識蒸留のようなトポロジーに対する仮定が少なく，局所的にも適用できるノードの枝刈りを対象とする．具体的には，非シグモイド活性化関数を扱うため，以前提案したノードエントロピーに基づくノード枝刈り法 [25] を拡張する．また，不均一なニューラルネットワークを扱うため，層間ペアリング，バイパス接続プルーニング，ネットワーク全体に対する枝刈り率設定に関する 3 つの規範を提案する．提案した方法と規範を TDNN (Time Delay Neural Network), TDNN-F (Factorized TDNN)，バイパス接続を有する，オープンソースの音声認識システム Kaldi の音響モデルに対して適用し検証を行った．

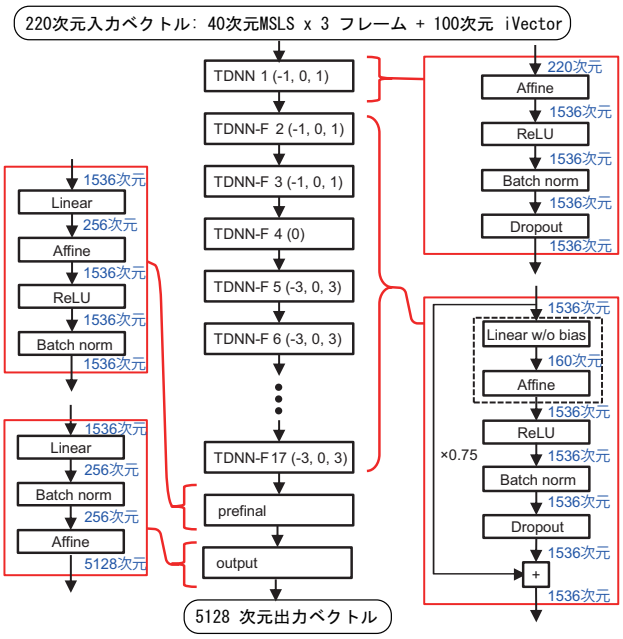


図 1: Kaldi の Nnet3-Chain 音響モデルの構成 (独自レシピ)

2 Kaldi のネットワーク構成

Kaldi の音響モデル学習には，日本語話し言葉コーパス (CSJ) レシピ [27] をベースとして，性能向上のために以下の変更を行った独自レシピを使用した．

- LibriSpeech レシピ [28] を参考に，nnet3-chain モデル [29] をサポートするための拡張．
- MFCC 特徴量の代わりに，40 次元 MSLS (Mel-Scale Log Spectrum) 特徴量 [30] を使用．これにより，入力には，3 フレーム分の MSLS 特徴量と 100 次元 iVector で構成される 220 次元のベクトルを使用．
- 言語モデル学習には，CSJ レシピで使用される SRILM [31] に代え，pocolm [32] を使用．

図 1 に，独自レシピで得られる全 19 層からなる音響モデルの構造を示す．第 1 層は TDNN で，フレームごとに 220 次元の入力ベクトルを受け取り，1,536 次元のベクトルを出力する．次の 16 層は TDNN-F で，各層は 160 次元のボトルネック層とバイパス接続を内包している．pre-final 層は バイパス接続はなく，256 次元のボトルネック層のみで構成される．出力層は，全結合ボトルネック層であり，5,128 次元ベクトルを出力する．バッチ正規化はすべての層で，ドロップアウトは最後の 2 層を除くすべての層で行われる．TDNN は，第 1 層から第 4 層目までは，連続 3 フレームを使用する．これを $(-1, 0, 1)$ と記述する．第 5 層では，現在のフレームのみが使用される．第 6 層から第 17 層目ま

では，高速な処理を実現しつつ，より長いコンテキストを扱うため，3 フレームおきに 3 フレーム分，つまり $(-3, 0, 3)$ が使用される．出力層を除き，活性化関数には，ReLU [26] が使用される．各層は，因子分解用の線形サブレイヤとアフィンサブレイヤ，ReLU 用の非線形サブレイヤ，バッチ正規化用サブレイヤ，ドロップアウト用サブレイヤなど 5 – 8 のサブレイヤで構成される．

3　提案するノード枝刈り手法

提案するノード枝刈り法，および，不均一なネットワークに枝刈りを適用するための規範を説明する．

3.1　ノード活性度の定義

l 番目の層の i 番目のノード $x_{l,i}$ に対するノードエントロピー q_e は，次式で定義できる [16, 25]．

$$q_e(l,i|D) = -\frac{N_0}{N_{0+1}} \log \frac{N_0}{N_{0+1}} - \frac{N_1}{N_{0+1}} \log \frac{N_1}{N_{0+1}}, \quad (1)$$

ここで，D はデータセット，N_0 と N_1 は，D に対して，閾値よりも低い値および高い値となったシグモイド活性化関数の出力数である．N_{0+1} は D 内のサンプル数であり，$N_0 + N_1$ と等しい．

ノードエントロピーは，ノードごとに異なる活性度を表現できるため，[16] らが提案している出力重みノルムよりもモデル圧縮性能がよいことが報告されている [25]．ただし，活性化関数としてシグモイドが前提であるため，N_0 と N_1 を判定するための閾値は，シグモイド関数の値域 (0–1) の中間値 0.5 であった．しかし，Kaldi では，値域が 0 から無限大である ReLU が活性化関数として用いられている．このため，閾値を 0.5 から 0 に近い ϵ に変更した．これは，閾値を単に変更したというよりも，ノードが活性化しているかどうかを活性化関数の出力の高低ではなく，0 かそれ以外かによって判断するように考え方を変更したと言える．この定義が有効かどうかを検証する意味で，比較のため，これに加え，頻度ベースのノード活性度 q_f と分散ベースのノード活性度 q_v という 2 つのノード活性度の定義を行った．

$$q_f(l,i|D) = \frac{N_0}{N_{0+1}}, \quad (2)$$

$$q_v(l,i|D) = \frac{1}{N_{0+1}} \sum_{t=1}^{N_{0+1}} x_t^2 - \bar{x}^2, \quad (3)$$

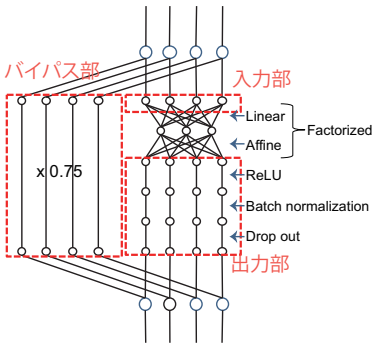


図 2: A TDNN-F layer

ここで，x_t は D における t 番目のサンプルの ReLU 出力値であり，$\bar{x}$ はデータセット内のすべてのサンプルの平均を示す．

3.2　ネットワークトポロジーの考慮

近年のニューラルネットワークのトポロジーは，TDNN，バイパス接続，attention モデルの登場により，加速度的に複雑化しており，層内でも複数のブロックで構成される不均一な構造となっている．図 2 は，Kaldi の典型的な TDNN-F 層の構造を示している．入力部，出力部，およびバイパス接続部と 3 つのブロックからなる複雑なトポロジーを有していることがわかる．

まず，出力部について考える．出力部は，ReLU サブレイヤーが含まれている．このため，ReLU の直後 (バッチ正規化の前) の値を用いて，式 (1) の ReLU バージョンに基づき，各ノードのノード活性度を求める．得られるノード活性度の小さいものから，枝刈りを行う．枝刈り率が与えられた際の具体的な枝刈り法については，後述する．

次に入力部について考える．図 2 のボトルネック部のみに注目すると，出力部とは独立して枝刈りができるように見える．しかし，実際には，入力部の入力は前の層の出力部に直接，接続されている．このため，前の層の出力部で枝刈りされたノードに直接接続されている入力部のノードを，枝刈りするものとした．これを「層間ペアリング」と呼ぶことにする．一方で，同じ層内の入力部と出力部はバイパス部を介して接続されている．この点に注目すれば，同じ層内の対応する入力部と出力部のノードを一緒に枝刈りする必要があると考えることもできる．これを「層内ペアリング」と呼ぶことにする．評価実験では，入力部と出力部を独立に枝刈りする場合を加えて，提案する層間ペアリングが良好な結果となるか比較実験を行うものとする．

バイパス部については，ResNet [33] に代表されるように，勾配消失を回避するのに役立つため，入力部と出力部の枝刈り結果に関係なく，枝刈りを行わないものとした．入力部，または出力部，あるいはその両方に

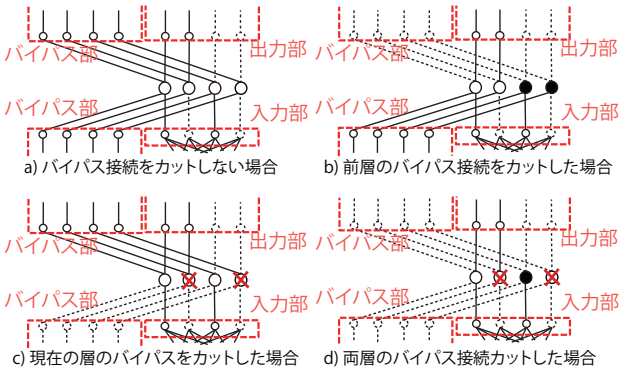
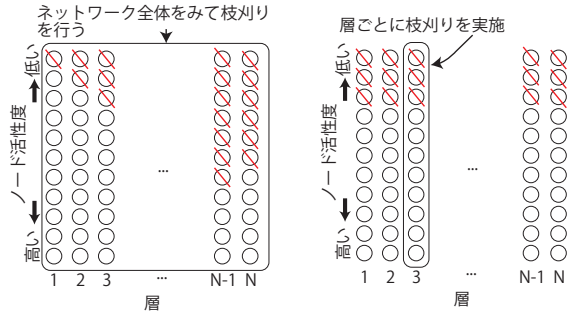


図 3: 層間接続問題．点線と点線の丸は枝刈りされたことを示す．層間の接続を保つため，上下の層の枝刈りの状況に応じて，常に 0 を出力するノード (黒丸) や終端ノード (赤×印) を設ける．



a) ネットワークベース　　b) 層ベース

図 4: 2 種類の枝刈りポリシー． 枝刈り率 30% の例:「ネットワークベース」は，ネットワーク全体を考慮して，活性度の小さいノードから順に 30% を枝刈りする．一方,「層ベース」は，層ごとに，活性度の小さいノード 30% を枝刈りする．

対応するノードが枝刈りされた場合に，バイパス接続を枝刈りの是非は，評価で比較実験を行うものとする．

枝刈り後，図 3 に示すように，入力部，出力部，バイパス部の接続を見ると，どちらか一方のノードだけが枝刈りされる接続がある．このようなオープンエンドの接続を扱うため，[33] に倣い，図 3b) および d) に示すように常にゼロ出力を行うノード (黒丸) や図 3c) および d) に示すよう終端ノード (赤×印) を設けるものとした．

以上，一つの層に対する枝刈りについて考察したが，実際のネットワークは多層に渡るため，枝刈り率[1]が与えられた際に，ネットワーク全体を考慮して枝刈りポリシーを決める必要がある．大きく 2 つのポリシーが考えられる．一つは，図 4a) に示すような「ネットワークベース枝刈り」ポリシーである．これは，層ごとの枝刈り率は一定でなくてよく，ネットワーク全体で平均して目的の枝刈り率になっていればよいとするポリシーである．実際，強化学習の結果として，枝刈り率は層ごと変えた方がよいとする報告もある [34]．また，以下の 3 点が冒頭に述べた過剰パラメータ化の知見として報告されている [1, 2, 3]．

1. 学習後，複数の中間層を初期化しても性能は維持される．
2. 入力層，または出力層が，1〜2 エポック後の対応する層と置き換えられると，性能が低下する．
3. 一般に，上位層はネットワークに対する貢献度が低くなる傾向がある．

これらの知見は，層ごとに異なる枝刈り率を許す，つまりネットワークベース枝刈りポリシーを支持しているといえる．

一方で，我々は，これに反して，図 4b) に示すように，すべての層で同じ枝刈り率となる必要がある「層ベース枝刈り」ポリシーを用いることを提案する．この背景となる考え方は，以下の通りである．

1. 上位層は下位層よりも出力層に近いため，最終出力により寄与するべきである．
2. ネットワークベース枝刈り率を適用するには，異なる層のノード間の活性度の違いを公正に評価する必要があるが，これは困難であるため，層ベースの枝刈り率で十分である．
3. ネットワークベース枝刈り率を適用すると，上位層の貢献度が低くなるため，積極的に上位層のノードが枝刈りされる．モデルの圧縮率を上げるには，枝刈り率を高くする必要があるが，この場合，上位層のノード数が少なくなり，モデルを十分表現できなくなる可能性がある．

これらの枝刈りポリシーについても評価実験で比較実験を行う．

表 1: **C1** – **C5** の実験条件．提案手法は太字

実験	ノード活性度	枝刈り率	入出力ペアリング	バイパス枝刈り	枝刈りポリシー
C1	**エントロピー**	0-70%	出力部のみ	**なし**	**層ベース**
	頻度				
	分散				
	ランダム				
C2	**エントロピー**	50%	**層間**	**なし**	**層ベース**
			層内		
			独立		
			出力部のみ		
C3	ランダム	0-70%	**層間**	**なし**	**層ベース**
				あり	
C4	**エントロピー**	0-70%	出力部のみ	**なし**	**層ベース**
					ネットワークベース
C5	**エントロピー**	50%	**層間**	**なし**	**層ベース**
	N/A	0%	N/A	**なし**	N/A

[1]本稿では，枝刈り率は，出力部内のノード総数と枝刈りされたノード数の比として定義するものとする．

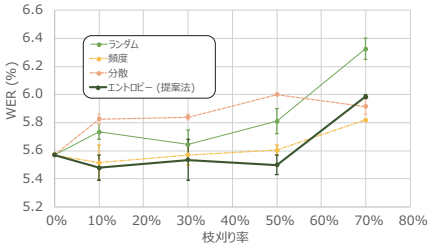


図 5: **C1**:ノード活性度比較結果 (JNAS)

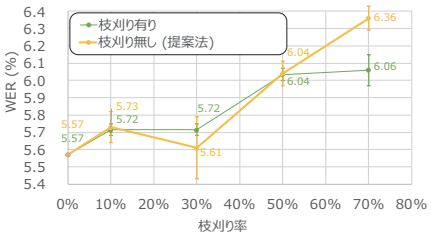


図 7: **C3**: バイパス部枝刈り比較結果 (JNAS)

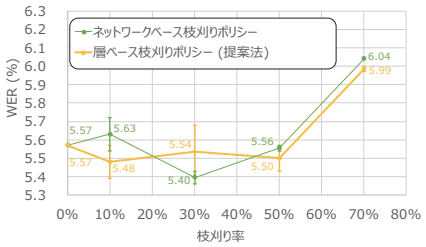


図 9: **C4**:枝刈りポリシー比較結果 (JNAS)

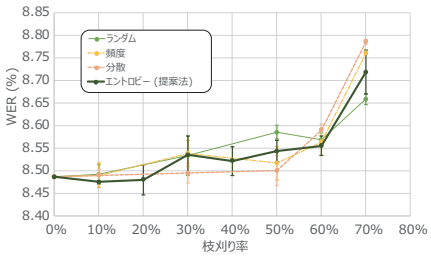


図 6: **C1**:ノード活性度比較結果 (CSJ)

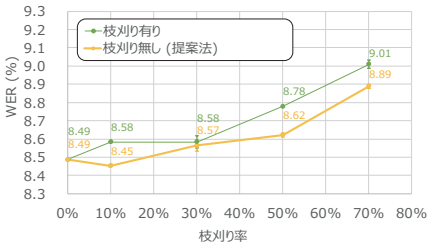


図 8: **C3**: バイパス部枝刈り比較結果 (CSJ)

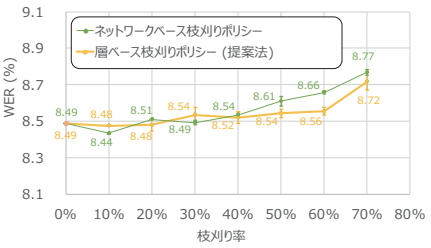


図 10: **C4**:枝刈りポリシー比較結果 (CSJ)

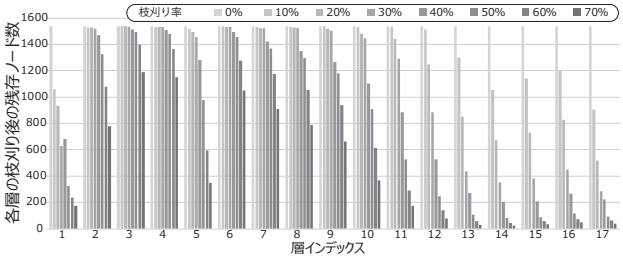


図 11: ネットワークベース枝刈りポリシー適用後の各層のノード数．層ベース枝刈りポリシー適用後は，各層のノード数はすべて同数．

表 2: **C2**:入出力間ペアリング比較結果．枝刈り率 50%．

コーパス	出力部のみ (上限値)	独立	層内	層間 (提案法)
JNAS	5.50	5.72	5.88	**5.70**
CSJ	8.54	8.69	8.61	**8.56**

表 3: 音声認識速度と WER．提案枝刈り法は，ノードエントロピーベースのノード活性度推定，層間ペアリング，バイパス接続の枝刈りなし，層ベース枝刈りポリシーの組合せ．

	音声認識速度 (s)	WER (%)
枝刈りなし	3,036	8.49
提案枝刈り法	2,388 (31% up)	8.57

4 評価

提案手法の有効性を検証するため，**C1** – **C5** の 5 種類の実験を行った．評価指標には，単語誤り率 (WER) を使用した．**C5** では，併せて音声認識システムの速度の計測も行った．表 1 に詳細な実験条件を示す．

C1: ノードエントロピーベースのノード活性度の検証．「ノードエントロピー (提案法)」,「頻度ベース」,「分散ベース」,「ランダム」の 4 種類のノード活性度を比較した．他の条件は以下の通り．枝刈り率は 0%，10%，30%，50%，60%，70%．枝刈りは出力部に対してのみ実施．バイパス部と入力部は，枝刈りなし．層ベース枝刈りポリシーを使用．

C2: 入出力層のペアリング基準の検証．「層間ペアリング (提案法)」と「層内ペアリング」に加えて，「独立」と「出力部のみ (オラクル)」の 2 つも比較に加えた．「独立」は，入力部のノードを出力部と無関係にランダムに枝刈りすることを意味する．「出力部のみ」は，入力部の枝刈りを行わないため，**C1** と同じ条件となり，原理上の性能上限値と言える． なお，**C1** の結果から，枝刈り率 50%までは同等の WER が得られることが判明したため，枝刈り率は 50% に固定した (層ベース枝刈りポリシー)．

C3: バイパス部の枝刈り是非の検証．バイパス接続の枝刈りを行う場合と行わない場合を比較した．層ベース枝刈りポリシーを用いて，枝刈り率 0 –

70% を検証．ノード活性度の定義とは無関係にこの要因を調査するため，出力部の枝刈りにおけるノード活性度には **C1** のランダム条件を用いた．また，入出力間ペアリングには層間ペアリングを用いた．

C4: 枝刈りポリシーの検証．「層ベース (提案法)」と「ネットワークベース」の枝刈りポリシーを比較した．枝刈り率は 0%–70%に変更した．また，ノードエントロピーベースの枝刈りと層間ペアリングを使用した．

C5: 上記の 4 提案手法，つまり，エントロピーベースのノード活性度，層間ペアリング，バイパス接続枝刈りなし，層ベース枝刈りポリシーを組み合わせた提案手法による音声認識速度向上の検証．枝刈り率を 50% に設定し，枝刈りなしの場合と比較を行った．

4.1 実験条件

学習とテスト用のデータセットには，JNAS (新聞記事読み上げ音声コーパス) と CSJ (日本語話し言葉コーパス) の 2 種類のサイズの異なる日本語コーパスを用いた．JNAS [35] は，60 時間の学習用音声データセットと，男女それぞれ 23 名，50 文からなるテストデータセットからなる小規模のコーパスである．CSJ [36] の学習用音声データセットは，JNAS の 10 倍強の 660 時間の音声データからなっている．また，それぞれ約 1,300 の発話からなる 3 種類のテスト用データセット (eval1–3) を含んでいる．テスト用データセットの総発話数は，3,949 で，計 18,376 秒である．式 (1)–(3) でノード活性度を求めるためのデータセット D は，すべての実験で共通の 300 発話 (19 分) を使用した．式 (1) および式 (2) の，ϵ は 0.001 に設定した．また，いずれの実験でもノード枝刈りを行った後，1 エポック分の再学習を行った． 実験は同じ条件下で 2 回行い，平均 WER を計算した．CSJ では，3 つのテストデータセットの平均 WER として算出した．

4.2 結果

図 5 と図 6 に，JNAS と CSJ に対する **C1** の結果を示す．図の横軸と縦軸は，それぞれ枝刈り率と WER を示す．JNAS では，ノードエントロピーベースのノード活性度が最もよい性能を示した．また，頻度もランダムを上回り，有効であることがわかる．CSJ では，ノードエントロピーは，枝刈り率が 0% から 60% まではランダムに上回り，枝刈り率変化に対して最も安定した性能を示したが，3 つの手法に明確な違いは認められなかった．全体としては，提案するノードエントロピーが最もよい性能を示したといえる．これは，ノード活性度を求める際に，ReLU の閾値として *epsilon* を使用するという考え方が正しいことを指示するといえる．提案したノードエントロピーベースのノード活性度が，他よりも基本的に良好な結果となったという事実は，1）分散ベースの方法は，ガウス分布を想定しているのに対し，ReLU の出力は非線形かつ非対称な分布を持っているため，2）頻度ベースの方法（式 (2)）は，式 (1) の右辺の最初の項のみを考慮しており，第 2 項を考慮していないため，ということを考慮することによって説明できよう．

表 2 に，枝刈り率 50% 時の JNAS と CSJ に対する **C2** の結果を示す．「出力のみ」は **C1** と同じ条件であるため，JNAS と CSJ の結果は，図 5 と図 6 の 枝刈り率 50% のエントロピーベースの結果と同じである．「出力のみ」は入力部の枝刈りを行わないため，原理的な上限値である．入力部の 3 つの枝刈り条件の中では，提案手法である層間ペアリングは「出力のみ」と同等の性能となり，最もよい性能を示した．これは，対象としている層の入力部が前の層の出力部と直接接続されているのに対し，同じ層の出力部とはボトルネック層を通じて間接的に接続されていることを考慮すると，直感的に理解できよう．

図 7 と図 8 に，**C3** の JNAS と CSJ に対する結果を示す．横軸と縦軸は枝刈り率と WER を表す．JNAS では，いずれも同等の性能となったが，CSJ では，入力部や出力部のノードが枝刈りされていた場合でも，対応するバイパス接続は枝刈りせず残したほうが良好な性能が得られた．これらの結果を考慮すると，バイパス接続は枝刈りしない方がよいと考えられる．これは，バイパス接続がネットワーク内の他の部分と比較して，音声認識性能に大きく貢献していることを示唆している．

図 9 と図 10 に，**C4** の JNAS と CSJ に対する結果を示す．横軸と縦軸は，それぞれ枝刈り率と WER を表す．枝刈り率が 40%未満の場合，JNAS と CSJ 共に，枝刈りポリシーの違いで性能の差は見られなかった．しかし，枝刈り率が 50% を超えると，提案する層ベース枝刈りポリシーが優位となった．これを分析するために，図 11 に，ネットワークベース枝刈りポリシーを適用した際に各層で残ったノード数を示す．上位層になるほど枝刈り率が高くなっていること，そのために上位層のノード数が少なくなっていることがわかる．つまり，枝刈り率が高いと，上位層のノード数が少なくなりすぎ，性能が劣化している可能性があることを示している．一方，層ベース枝刈りポリシーを適用した場合は，残ったノード数はどの層でも同数に保たれるため，枝刈り率が高くても性能が安定していると考えられる．

表 3 に，**C5** の結果を示す．テストデータには，18,376 秒の音声データからなる CSJ テストデータセット (eval1-

3) を使用した．音声認識の処理速度は，音響モデルの尤度計算部だけではなく，音声認識システム全体の処理時間として，Intel(R) Xeon(R) E5-2697A v4 (2.6 GHz) の 1 コアを用いて計測した．枝刈りを行わない場合，処理時間は 3,036 秒で，平均 WER は 8.49% であった．提案する枝刈り手法では，枝刈り率 50% で，それぞれ 2,388 秒と 8.57% であった．つまり，音声認識性能を維持しながら，31% の速度向上が達成することができた．

C1–**C5** の結果から，エントロピーベースのノード活性度推定，層間ペアリング，バイパス部の枝刈りをしない，層ベース枝刈りポリシーの組み合わせである提案法が最もよい性能となることを示すことができた．**C1**，**C3**，および **C4** の結果から 2 つのサイズが異なるコーパスである JNAS および CSJ ともに，枝刈り率は 50% が適切であることが得られ，これによって，**C5** で 31% の速度向上を達成した．WER と音声認識の処理速度のバランスは，アプリケーションによって変わる可能性がある．提案法は，JNAS のようにモデル圧縮と過学習しやすい小規模なコーパスだけでなく，CSJ のように規模の大きいコーパスでも効果的であった．これは，ネットワークトポロジを考慮して枝刈りを行うことが重要であることを示している．また，過剰パラメータ化 [1, 2, 3] の知見から得られる予測と異なり，層ベース枝刈りポリシーが良い結果を示した．これは，証明には，さらなる研究が必要なものの，3.2 節で述べた背景となるアイデアが正しいことを示唆している．

5　おわりに

本稿では，複雑で不均一なニューラルネットワークのモデル圧縮手法を扱った．このために，ノードエントロピーベースのノード活性度推定法，層間ペアリング，バイパス接続の枝刈りをしない，層ベース枝刈りポリシーという 4 つの手法と規範からなるノード枝刈り方法を提案した．提案法を Kaldi の音響モデルに適用し，その有効性を明らかにした． より大きなコーパス，多言語での有効性検証，ニューラルアーキテクチャ探索を使用した自動パラメータ推定，アテンションモデルなどより複雑なネットワークへの適用が今後の課題である．

謝辞

HRI-JP の 尾西ダニーロ，瀧ケ平 雅行，住田 直亮，中塚 雅樹の各氏に感謝する．

参考文献

[1] M. Belkin, D. Hsu, S. Ma, and S. Mandal, “Reconciling modern machine-learning practice and the classical bias–variance trade-off,” *Proc. of the National Academy of Sciences*, vol. 116, no. 32, pp. 15 849–15 854, 2019.

[2] C. Zhang, S. Bengio, and Y. Singer, “Are all layers created equal?” *CoRR*, vol. abs/1902.01996, 2019. [Online]. Available: http://arxiv.org/abs/1902.01996

[3] C. Zhang, S. Bengio, M. Hardt, M. C. Mozer, and Y. Singer, “Identity crisis: Memorization and generalization under extreme overparameterization,” in *Proc. of 8th International Conference on Learning Representations (ICLR)*, 2020. [Online]. Available: https://openreview.net/forum?id=B1l6y0VFPr

[4] Y. Le Cun, J. S. Denker, and S. A. Solla, “Optimal brain damage,” in *Proc. of the 2nd International Conference on Neural Information Processing Systems (NIPS)*. Cambridge, MA, USA: MIT Press, 1989, p. 598–605.

[5] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, “A survey of model compression and acceleration for deep neural networks,” *CoRR*, vol. abs/1710.09282, 2017. [Online]. Available: http://arxiv.org/abs/1710.09282

[6] J. Xue, J. Li, and Y. Gong, “Restructuring of deep neural network acoustic models with singular value decomposition,” in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH 2013)*, 2013, pp. 2365–2369.

[7] 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 *Proc. of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6655–6659.

[8] Y. Gong, L. Liu, M. Yang, and L. D. Bourdev, “Compressing deep convolutional networks using vector quantization,” *CoRR*, vol. abs/1412.6115, 2014. [Online]. Available: http://arxiv.org/abs/1412.6115

[9] V. Sindhwani, T. Sainath, and S. Kumar, “Structured transforms for small-footprint deep learning,” in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 3088–3096.

[10] M. Pöllot, R. Zhang, and A. Kaup, “An efficient alternative to network pruning through ensemble learning,” in *Proc. of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4022–4026.

[11] J. Ba and R. Caruana, “Do deep nets really need to be deep?” in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, 2014, pp. 2654–2662. [Online]. Available: http://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep

[12] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in *NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[13] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size dnn with output-distribution-based criteria," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH 2014)*, 2014.

[14] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," in *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016. [Online]. Available: http://dx.doi.org/10.18653/v1/D16-1139

[15] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, "Revisiting knowledge distillation via label smoothing regularization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 3903–3911.

[16] T. He, Y. Fan, Y. Qian, T. Tan, and K. Yu, "Reshaping deep neural network for fast decoding by node-pruning," in *Proc. of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 245–249.

[17] T. Ochiai, S. Matsuda, H. Watanabe, and S. Katagiri, "Automatic node selection for deep neural networks using group lasso regularization," in *Proc. of 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5485–5489.

[18] R. Takeda, K. Nakadai, and K. Komatani, "Acoustic model training based on node-wise weight boundary model for fast and small-footprint deep neural networks," *Computer Speech & Language*, vol. 46, pp. 461 – 480, 2017.

[19] 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 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 1135–1143.

[20] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen, "Compressing convolutional neural networks," *CoRR*, vol. abs/1506.04449, 2015. [Online]. Available: http://arxiv.org/abs/1506.04449

[21] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *Proc. of 4th International Conference on Learning Representations (ICLR)*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1510.00149

[22] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in *Proc. of 5th International Conference on Learning Representations (ICLR)*, 2017. [Online]. Available: https://openreview.net/forum?id=rJqFGTslg

[23] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," in *Proc. of 5th International Conference on Learning Representations (ICLR)*, 2017. [Online]. Available: https://openreview.net/forum?id=SJGCiw5gl

[24] A. See, M.-T. Luong, and C. D. Manning, "Compression of neural machine translation models via pruning," in *Proc. of The 20th SIGNLL Conference on Computational Natural Language Learning*. ACL, 2016. [Online]. Available: http://dx.doi.org/10.18653/v1/K16-1029

[25] R. Takeda, K. Nakadai, and K. Komatani, "Node pruning based on entropy of weights and node activity for small-footprint acoustic model based on deep neural networks," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH 2017)*, 2017, pp. 1636–1640.

[26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. of the 27th International Conference on Machine Learning (ICML'10)*, 2010, p. 807–814.

[27] "Kaldi CSJ recipe (confirmed on Aug. 14, 2020)," https://github.com/kaldi-asr/kaldi/tree/master/egs/csj/s5/local/chain/tuning/run_tdnn_1a.sh.

[28] "Kaldi liblispeech recipe (confirmed on Aug. 14, 2020)," https://github.com/kaldi-asr/kaldi/tree/master/egs/librispeech/s5/local/chain/tuning.

[29] "Kaldi nnet3-chain model (confirmed on Aug. 14, 2020)," https://kaldi-asr.org/doc/chain.html.

[30] Y. Nishimura, T. Shinozaki, K. Iwano, and S. Furui, "Noise-robust speech recognition using multi-band spectral features," in *Proc. of 148th Acoustical Society of America Meetings*, no. 1aSC7, 2004.

[31] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "Srilm at sixteen: Update and outlook," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE SPS, 2011.

[32] "Pocolm web (confirmed on Aug. 14, 2020)," https://github.com/danpovey/pocolm.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[34] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "Amc: Automl for model compression and acceleration on mobile devices," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2018, pp. 784–800.

[35] K. Itou, M. Yamamoto, K. Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano, and S. Itahashi, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research," *Journal of the Acoustical Society of Japan*, vol. 20, no. 3, pp. 199–206, 1999.

[36] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," in *Proc. of the Second International Conference on Language Resources and Evaluation (LREC'00)*. ELRA, 2000. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2000/pdf/262.pdf