Vol.2019-SLP-130 No.2
2019/12/6

CNNフロントエンドによる End-to-End連続DPマッチングの高速化

田中 智宏[1]　篠崎 隆宏[1]

概要：我々はこれまでに単語境界が未知の条件で任意のキーワードを検出する連続 DP マッチングを 2D-RNN を用いた End-to-End 型のニューラルネットとして実現する手法を提案し，従来の音素事後確率を用いた連続 DP や単語埋め込み法を上回る検出性能を示した．しかし，2D-RNN の展開後のサイズがキーワードテンプレートと入力の長さの積に比例するため，計算量が非常に大きいという問題があった．そこで今回 CNN フロントエンドを用いて End-to-End 連続 DP の計算量を CNN カーネルの幅と長さの積および CNN 階層数の指数オーダーで削減する手法を提案する．雑音存在下のキーワード検出実験において，提案法は従来の 2D-RNN 型連続 DP の検出性能を保ちつつ，計算速度を大幅に改善することを示す．

1. はじめに

2D-RNN 型連続 DP [1] はキーワードオープンな条件での連続単語検出においてすでに高い性能を示していたが，計算量が大きいという問題があった．我々はこれを CNN を用いて高速化する手法を提案する．提案法は CNN の時間方向の縮小倍率と同じ倍率の高速化が可能である．実験ではストライド 3 の畳み込み層 2 層からなる CNN を用い，性能をほぼ維持しながらおよそ 9 倍の高速化が可能であることを示す．また，近年盛んに研究が行われている埋め込みベースの手法 [2], [3], [4] のうち，2D-RNN 型連続 DP と同じく始終端条件のないモデル [1] と比較し，キーワードオープンな条件における検出性能が優れていることを示す．

2. 提案モデル

提案モデルの全体構造を図 1 に示す．この連続 DP モデルが取り組むタスクはキーワード検出であり，テンプレート音声を用いて検索対象の音声からキーワードが発話されたことを検知するものである．ここで，検索対象の音声を $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_T\}$，テンプレート音声を $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_M\}$ とする．また，キーワードが発話されたタイミングで検知するためには，発話が終了したフレームを予測する必要がある．この，各フレームで発話が終了した事後確率の予測を $\mathbf{y} = \{y_1, y_2, ..., y_T\}$ とする．本タスクの目的は，この $\mathbf{S}$ と $\mathbf{Q}$ から正しい $\mathbf{y}$ を求めることである．

従来の 2D-RNN 型連続 DP では，両音声 $\mathbf{S}, \mathbf{Q}$ は式 1 のように，特徴抽出器 f_{ext} によりフレームごとに特徴抽出がされたあと，その間の距離が距離関数 d により計算され距離行列 $\mathbf{D}$ を構成する．これを用いて式 2 のように DP マッチングを行い，その後式 3 のように事後処理 f_{post} をして予測事後確率 $\mathbf{y}$ を求める．

$$D_{t,m} = d\left(f_{ext}\left(\mathbf{s}_t\right), f_{ext}\left(\mathbf{q}_m\right)\right), \tag{1}$$

$$\mathbf{h}_{t,m} = f_{DP}\left(\mathbf{h}_{t-1,m}, \mathbf{h}_{t-1,m-1}, \mathbf{h}_{t,m-1}, D_{t,m}\right), \tag{2}$$

$$y_t = f_{post}\left(\mathbf{h}_{t,M}\right), \tag{3}$$

$$(t = 1, ..., T; m = 1, ..., M).$$

ここで，$f_{ext}, f_{DP}, f_{post}$ はニューラルネットにより実装され，特に f_{DP} は 2D-RNN により実装される．この DP 部分の計算量は O(MT) であり，並列計算する場合は $\mathbf{h}_{t-1,m}, \mathbf{h}_{t-1,m-1}, \mathbf{h}_{t,m-1}$ が求まらないと $\mathbf{h}_{t,m}$ が計算できないため，図 2 のように斜めの一列ずつ計算が進んでいくことになり，時間計算量は O(T)，ハードウェアコストは O(M) となる [5].

今回の提案は，DP マッチングを行う前に式 4 のように CNN を適用する手法である．これに対して DP マッチング（式 5）を行った後，式 6 のように事後処理時にフレームレートを元に戻す．

$$\mathbf{D}' = CNN\left(\mathbf{D}\right), \mathbf{D}' \in R^{T' \times M' \times C}, \tag{4}$$

$$\mathbf{h}'_{t',m'} = f_{DP}\left(\mathbf{h}'_{t'-1,m'}, \mathbf{h}'_{t'-1,m'-1}, \mathbf{h}'_{t',m'-1}, \mathbf{D}'_{t',m'}\right), \tag{5}$$

[1] 東京工業大学
Tokyo Institute of Technology, Tokyo, Japan
www.ts.ip.titech.ac.jp

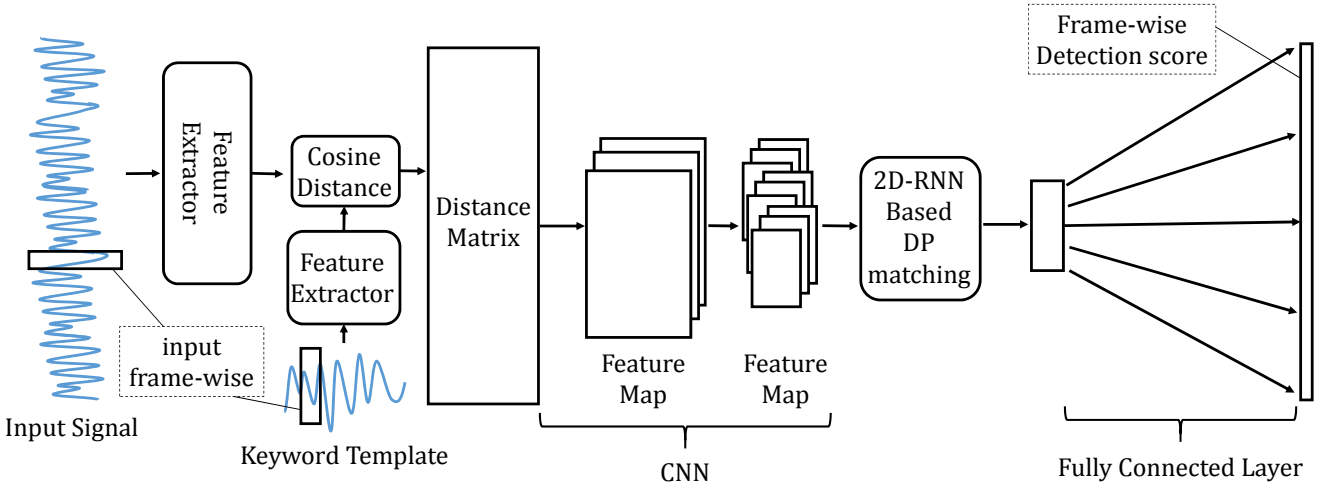

図 1　CNN フロントエンドを組み合わせた 2D-RNN 型連続 DP の全体構造

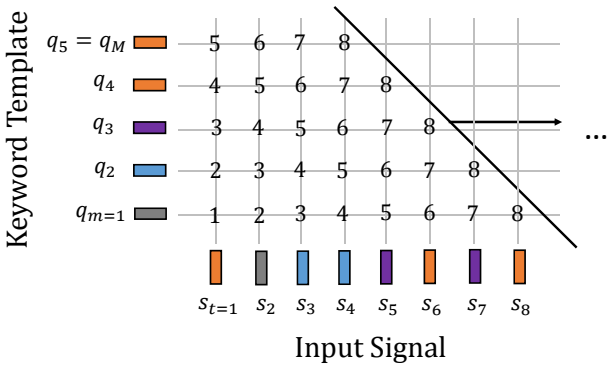

図 2　連続 DP の並列計算．番号 i が記されたノードは番号 i-1 が記されたノードの計算が終わったあと，同時に計算される．

$$[y_{kt'-k+1}, ..., y_{kt'}] = f_{post}\left(\mathbf{h}'_{t',M'}\right), k = T/T', \quad (6)$$
$$(t' = 1, ..., T'; m' = 1, ..., M').$$

このようにして T と M を $1/k$ に縮小した後で DP マッチングを行うことで，並列計算時の時間計算量を O(T) から $O(T') = O(T/k)$ に削減することができる．同時に，ハードウェアコストも O(M) から $O(M') = O(M/k)$ に削減できる．また，CNN で特徴抽出を行うことで，精度も維持されることが期待できる．

3. 実験条件

実験では Google Speech Commands Dataset [6] の音声コマンドデータ，および CHiME-3 [7] のノイズデータを用いた．音声コマンドデータは Kaldi toolkit [8] を用いた強制アライメントにより前後の空白を取り除いた．このとき，333 サンプルがアライメントに失敗したため，残りの 105,496 サンプルを実験で用いた．音声コマンドはキーワードの種類によって表 1 のように分割し，学習セットとキーワードクローズな評価セットのために 24 種類，キーワードオープンな評価セットのために 10 種類用いた．話者は学習・評価セットで独立で，それぞれ 1,308 人である．検索対象の音声はノイズ音のランダムな位置にランダムな音声コマンドを埋め込むことにより作成した．このとき，キーワード区間における SN 比の平均値が 5，20dB となるようにノイズ音声を一律で調整した．学習データでは GPU を用いて効率よく学習するために，5 秒間のノイズ中にコマンド音声を 1 つ埋め込み，これとテンプレートとして用いるランダムに選択された別のコマンド音声をペアにして学習データを作成した．このときノイズ中の埋め込んだ音声のキーワードとテンプレートのキーワードが一致するものをキーワードごとに 1,000 ペア，一致しないものを 1,000 ペアずつ作成し，合計 48,000 ペアを学習セットとして用いた．評価データではより現実的な条件で検証するため，3 分間のノイズ中にテンプレートと一致するキーワードのランダムな音声を 1 つ，一致しないキーワードのランダムな音声を 4 つ，それぞれ重ならないようにランダムな位置に埋め込み，これとランダムなテンプレートをペアにすることでキーワードクローズ評価セットを 480 ペア，キーワードオープン評価セットを 500 ペア作成した．特徴量は 13 次元の MFCC を用いた．MFCC の抽出は Kaldi toolkit を用い，窓長 25ms，窓シフト長 10ms で行われた．また，cepstral means and variance normalization(CMVN) を適用した．

表 1　学習・評価セットのキーワード種類．

Dataset	Keywords
学習セット，キーワードクローズ評価セット	”Zero” to ”Nine”, ”Yes”, ”No”, ”Up”, ”Down”, ”Left”, ”Right”, ”On”, ”Off”, ”Stop”, ”Go”, ”Backward”, ”Forward”, ”Follow”, ”Learn”
キーワードオープン評価セット	”Bed”, ”Bird”, ”Cat”, ”Dog”, ”Happy”, ”House”, ”Marvin”, ”Sheila”, ”Tree”, ”Wow”

2DRNN 型連続 DP の実装は，f_{ext} に 3 層の全結合層（それぞれ 13，8，4 ユニット），f_{DP} に 16 ユニットの 2D-GRU，f_{post} に 1 層の全結合層を用いた．CNN はカーネルサイズ

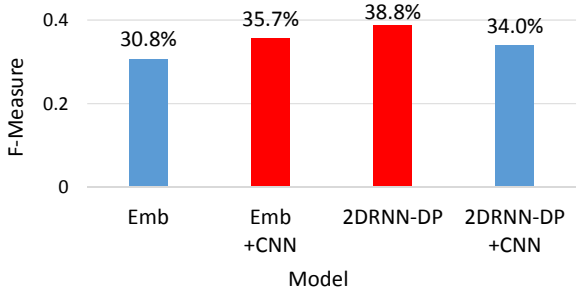

図 3 SN 比が 5dB，キーワードクローズな条件での F スコアの比較

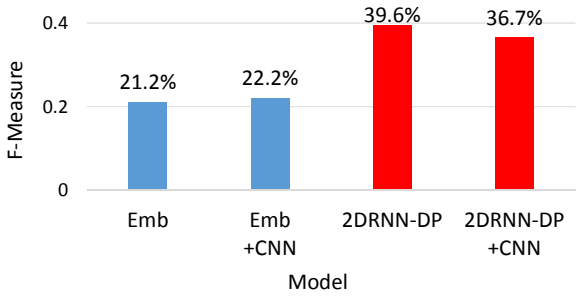

図 4 SN 比が 5dB，キーワードオープンな条件での F スコアの比較

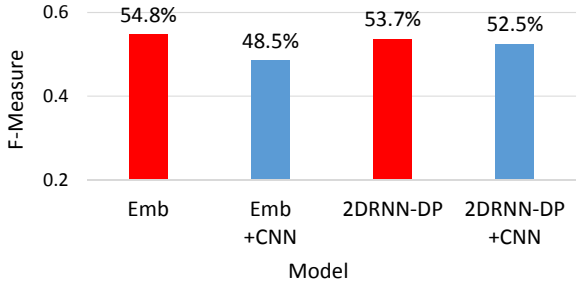

図 5 SN 比が 20dB，キーワードクローズな条件での F スコアの比較

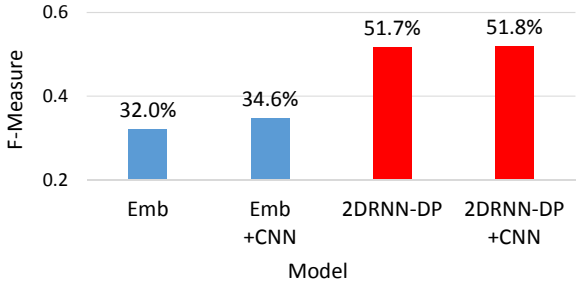

図 6 SN 比が 20dB，キーワードオープンな条件での F スコアの比較

とストライドが 3×3 の 2 層の畳み込み層を用い，カーネル数をそれぞれ 3 と 9 とした．また，CNN には Batch normalization [10] と dropout [11] を適用した．

ベースラインとして， [1] で提案された埋め込み型の連続キーワード検出器を用いた．CNN の効果を比較するため，この埋め込み型キーワード検出器の入力に CNN を適用したものも性能を求めた．以下では，これらのモデルをそれぞれ Emb，Emb+CNN と表す．また，2DRNN 型連続 DP モデルを 2DRNN-DP，これに CNN フロントエンドを組み合わせたモデルを 2DRNN-DP+CNN と表す．

目的関数にはソフト F 値 [1] を用い，Adam [9] により学習を行った．学習は異なる乱数のシード値を用いて 3 回ずつ行い，その中で開発セットにおいて最も良いモデルの性能を求めた．これをスプライシングの有無の両条件で行い，評価セットのスコアが高いほうを採用した．実際，2DRNN-DP，2DRNN-DP+CNN，および SN 比が 20dB のキーワードクローズな条件における Emb+CNN モデルはスプライシングありのほうが性能が高く，それ以外はスプライシングなしのほうが性能が高かった．

予測の正否は単語単位で行い，正しい単語終端を中心に手前に 9 フレーム，後ろに 10 フレーム許容した．単語終端の予測は予測事後確率が Φ_h を超えるフレームが Φ_w 回だけ連続して続いたフレームとした．閾値 Φ_h, Φ_w は学習セットとキーワード・話者クローズな開発セットを作成し，これを用いて決定した．正しい単語終端を求めるためのアライメントは Kaldi のトライフォン GMM-HMM モデルを用いた強制アライメントにより作成した．

表 2 3 分間のデータを処理するのに要した時間．100 回計測した平均値を記す．

Model	Eval. Time (sec)
Emb	18.07
Emb+CNN	2.06
2DRNN-DP	48.75
2DRNN-DP+CNN	5.38

4. 実験結果

計算時間の比較を表 2 に示す．たしかに CNN を適用することで理論上と同じ 9 倍ほど早くなったことがわかる．次に，キーワード検出精度の比較を図 3 から図 6 に示す．いずれの場合も 2DRNN-DP+CNN は 2DRNN-DP と同等か，絶対値で 1%から 5%ほどの悪化にとどまっていることがわかる．特に，キーワードオープンな条件においては，Emb モデルは性能が大幅に悪化する一方，2DRNN-DP は CNN を適用しない場合も含めて性能を維持していることがわかる．

5. まとめ

本研究では 2D-RNN 型連続 DP モデルに適用可能な CNN を用いた高速化手法を提案した．実験ではストライド 3 の畳み込み層 2 層からなる CNN を用い，従来法とほぼ同等な精度で約 9 倍の高速化ができることを示した．今後の課題としては，CNN モジュールを工夫することによる計算速度を維持したままでの更なる精度の向上や，実環境におけるデータを用いた評価実験などが挙げられる．

参考文献

[1] Tanaka, Tomohiro, and Takahiro Shinozaki. "F-Measure Based End-to-End Optimization of Neural Network Keyword Detectors." 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2018.

[2] Audhkhasi, Kartik, et al. "End-to-end ASR-free keyword search from speech." IEEE Journal of Selected Topics in Signal Processing 11.8 (2017): 1351-1359.

[3] Yuan, Yougen, et al. "Query-by-Example Speech Search using Recurrent Neural Acoustic Word Embeddings with Temporal Context." IEEE Access (2019).

[4] Chung, Yu-An, et al. "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder." arXiv preprint arXiv:1603.00982 (2016).

[5] Steffen, Peter, Robert Giegerich, and Mathieu Giraud. "GPU parallelization of algebraic dynamic programming." International Conference on Parallel Processing and Applied Mathematics. Springer, Berlin, Heidelberg, 2009.

[6] Warden, Pete. "Speech commands: A dataset for limited-vocabulary speech recognition." arXiv preprint arXiv:1804.03209 (2018).

[7] Barker, Jon, et al. "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines." 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, 2015.

[8] Povey, Daniel, et al. "The Kaldi speech recognition toolkit." IEEE 2011 workshop on automatic speech recognition and understanding. No. CONF. IEEE Signal Processing Society, 2011.

[9] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[10] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).

[11] Hinton, Geoffrey E., et al. "Improving neural networks by preventing co-adaptation of feature detectors." arXiv preprint arXiv:1207.0580 (2012).