

# 複雑なセンサ情報理解におけるディープラーニング

大北 剛<sup>1</sup> 井上 創造<sup>1</sup>

概要: 本論文は複雑なセンサ入力を扱うために, 組成セミマルコフ畳み込み LSTM モデルを導入する. この組成セミマルコフ畳み込み LSTM モデルは, 近年の CNN と RNN を複合したアーキテクチャをベースに, 組成インスタンスモデルとセミマルコフモデルという 2 つの機構を導入する. 実験結果が示すように, 組成インスタンスモデルを用いることにより, 27% から 43% に精度を向上させることができた.

## Deep Learning for Complex Sensor Information Understanding

Tsuyoshi Okita<sup>1</sup> Sozo Inoue<sup>1</sup>

### 1. イントロダクション

IoT において, センサネットワークの情報を人間の解釈できる形に翻訳すること (教師ありセンサ情報理解のタスクと呼ぶ) は重要な第一ステップとなる. このタスクはセンサ出力を入力として, これらを出力したと思われる単独もしくは複数のイベント/アクティビティを時系列で探知することを目的とする.

近年の深層学習の成功とともに, センサ出力の学習に深層学習を応用する例が見受けられるようになった. センサ出力を CNN (畳み込みニューラルネットワーク) により次元圧縮して分散表現し, さらにこれを RNN を用いて時系列に沿ったシグナルのパターンを分類するという形のアーキテクチャがいくつか提案されている. 深層センスモデル (DeepSense model)[12], 深層畳み込み LSTM モデル (Deep convolutional LSTM model)[7], 密なラベリングモデル (dense labeling)[13] などがそのような例である. これら 3 つはいずれも CNN (畳み込みニューラルネットワーク) と RNN を複合した形を取り, 最後に softmax で分類する. CNN による分散表現を用いることによりセンサフュージョン [10] と呼ばれる現象<sup>\*1</sup> を取り扱い易くし, 時間的なセンサ出力のパターンを RNN で捉えるというものである. 深層学習を部分的に適用して検証した小規模な実験, たとえば事前にセグメント化された行動のセンサ出力から RNN モデルが効果的であることを示した実験 [4] やセンサ出力と行動の分散表現の写像からまだ見ぬ行動を推定する実験 [6] も提案されている.

一つ目に, 複数のイベントがオーバーラップする状況は現実問題として頻出するものではないが, センサの種類に

よりそのような状況が起こっている可能性は否定できない. 前者 3 つの CNN と RNN の結合型のモデルにおいて, この状況はモデルとして扱ってはならず, センサ出力がこのような状況を捉えた場合, これらのモデルでは単に無視される. 無視してもこれらのモデルによる精度は高いため, データセットによってはあまりこういう事象を含まないものも考えられる. モデルによっては, たとえば, まだ見ぬ行動を推定するモデルにおいては, 複数の既出のイベントの結合したセンサ出力を新規の行動と誤って推定する可能性もある. この問題は組成モデル (compositional model) として機械学習で扱われることもある [5]. この組成モデルが機能すれば, 複数のイベントのオーバーラップのみならず, 各々のイベントが時間的, 空間的に拡大縮小される場合なども扱える可能性がある.

二つ目に, 各イベントの開始時刻, 終了時刻の推定 (セグメンテーションやスポッティングと呼ばれる) はセンサ情報を自動的に理解するには必須の技術である. semi-CRF モデル [11], 重点サンプリングを用いたベイズモデル [3] と同様, 上記 3 つの CNN+RNN モデルはセグメントとイベント/行動認識を同時に行なう手法であると言える. しかし, 行動認識や IoT のセンサ出力におけるイベント同士の関係はマルコフ連鎖ではなく, セミマルコフ連鎖である場合が多いことである. たとえば, スキップの動作に入る前の遷移状態として歩く動作が認識されるような場合はあるかもしれないが, 2 つのスキップの動作の間に明確な関係性はないのではと思われる. CNN+RNN モデルにおいて, RNN は時間的に後方のイベントに対して依存性を捉えるが, これを無視したい場合が多いのではと考える. Sarawagi and Cohen は固有表現認識器 (NER) や遺伝子探索などのような自然の事象において, セミ CRF を用いてこれを議論している [9].

<sup>1</sup> 九州工業大学

<sup>\*1</sup> センサデバイスが異なるために異なるセンサ出力を得る場合と異なる周波数で測定されたセンサ出力を用いる場合を含む.

## 2. 組成セミマルコフ畳み込み LSTM モデル

センサ情報を理解するアーキテクチャで、CNN と RNN を結合した形を取るものが 3 例提出されていることをイントロダクションで述べた。本論文ではこれらをベースとして、2つの拡張を行なう。一つ目は複数のセンサ出力が混合した場合である。上記のモデルのいずれにおいても、また多くの機械学習のモデルにおいても、特徴量が 2 倍になった場合に、それに対する分類結果としてコンポーネント 2 つを出力するモデルはあまり論じられていない。しかし、存在しないわけではなく、組成モデル (compositional model) として提案されている (Lloyd et al., 2014)。本論文においては、これを多層 RNN のコンテキストにおいて議論する。行動認識においては、疎な中にイベントが表われるという形が多く存在するため、この状況であれば、混合する状況に対処しても精度にはあまり表われない。しかし、もし対象とするデータ集合にセンサ出力の混合するものが存在すれば効果は大きく、そのような状況に対するアーキテクチャとして提案したい。二つ目は、セミ CRF [9] などで盛んに議論されてきたセミマルコフ連鎖に対する議論である。CNN と RNN を結合した 3 例において、この観点は議論されていない。そこで、これを少し議論したい。

### 2.1 組成インスタンスモデル

表 1 は、センサの出力が複数の源から発生することにより影響を受ける可能性があるかを概略したものである。但し、センサの実装の仕方によりこれは大きく影響を受けること、また、センサを実際にどう人間に装着させるか/固定物に固定させるか/その他などにより大きく異なる場合もある。しかし、ここで言えることは、センサ出力の源は単数と仮定されることが多いが、複数の源からの発生は十分にありえる設定だと言うことである。

このような前置きの下本論文の中心的話題に入ると、特徴量のいずれかにおいて、その根源となるインスタンスが複数である可能性がある場合に、分類結果としての出力を複数のラベルとできるフレームワークを作りたいというものである。「複数 (マルチ)」という言葉の連想から対応する機械学習のアルゴリズムがいくつかある。たとえば、マルチプルインスタンス学習 (multiple instance learning) である。マルチプルインスタンス学習においては、正のバッグ (bag)、負のバッグを与えられ (正のバッグと負のバッグはそれぞれ一つ以上のインスタンスを成分とする)、ラベルはインスタンスに与えられるのではなく、正のバッグ、負のバッグに与えられた状況において、分類学習を行なう。我々の対象としたい学習の方式は、特徴 (つまりセンサ出力) に応じて、ラベルが単数の場合と複数の場合が生じるという分類学習であるので、これとは若干異なる。組成モデル

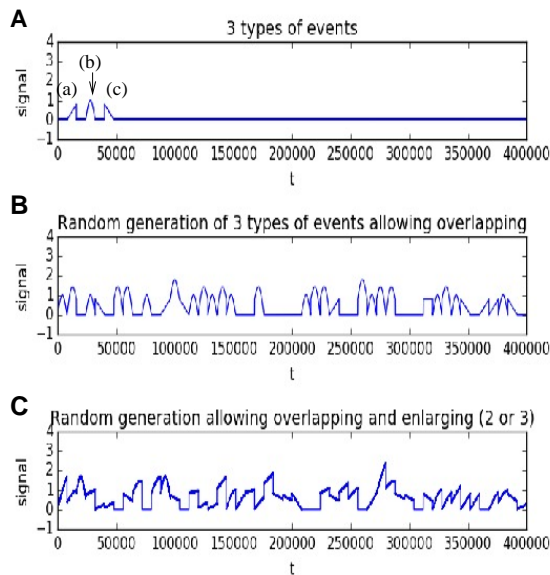


図 1 図 A は一例として、単純な 3 つのクラスのイベントを示す (これを直接トレーニング集合としない)。図 B と図 C は 2 種類の人工データを示す。図 B はクラスが重なって生成されることを許した場合、図 C はさらに時間軸に対して 2 倍や 3 倍に拡大することを許した場合のセンサ出力の一例を示す。

小分類	代表的なセンサ	混合可能性
化学センサ		
ガスセンサ	可燃性ガスセンサ、NOx センサ、酸素センサ、水素センサ	○
においセンサ	においセンサ、口臭センサ、悪臭センサ	○
湿度センサ	相対湿度センサ、絶対湿度センサ、結露センサ	○
イオンセンサ	pH センサ、Na イオンセンサ、K イオンセンサ	○
味覚センサ	甘味センサ、味覚センサ、塩味センサ、苦味センサ	○
バイオセンサ	免疫センサ、酵素センサ、微生物センサ	○
物理センサ		
光センサ	イメージセンサ、赤外線センサ、カラーセンサ	△
温度センサ	温度センサ、放射温度計	△
圧力センサ	絶対圧センサ、ゲージ圧センサ	△
速度センサ	速度センサ、流速センサ、ドップラーセンサ	△
加速度センサ	加速度センサ、ノックセンサ	△
超音波センサ	バイモルフ超音波センサ、音電型超音波センサ	△
磁気センサ	磁気抵抗素子、ホールセンサ	△
電解センサ	電解センサ、電圧センサ	△
放射線センサ	α 線センサ、β 線センサ、γ 線センサ	○

表 1 センサの感知する物理量や化学量を感じた場合に源が分解する可能性があるか。○は可能、×は不可能、△はこれらの中 (状況に依存) とする。

(compositional model) という形の学習が存在し、回帰モデルにおいて適用例がある (Lloyd et al., 2014). 複数の回帰モデルを組み合わせることにより最終的な出力を決めるため、学習の際にはある出力が混合したものであることを判断し、混合したものであれば各々の回帰モデルを同定する。したがって、基本的な枠組みには類似点がある。しかし、相違点もある。Lloyd et al. の組成モデルにおいては、モデルをどう混合するかを時系列に沿って推定が行なわれる。どう混合されるかをマルチプルカーネル学習などを適用して統計的に学習し、これから全体の組成モデルを学習する。そして、この組成モデルを学習することにより各コンポーネントを分離する。われわれの考えたい状況においては、複数の分類タスク間には統計的な学習はその場その場によるため統計的な規則は存在せず、したがってこの規則を学習することはできない。またそのような規則を学習したいわけではない。このような類似性と相違点より、われわれはこのモデルを組成モデルではなく、(これ以降) 組成インスタンスモデルと呼びたい\*2。

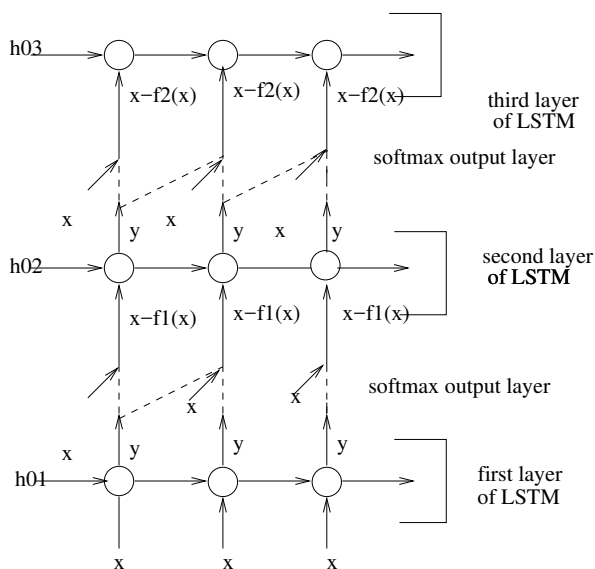


図 2

図 2 は、以上を LSTM で実装する場合のアイデアを示す。例えば 3 つのコンポーネントまで認識できる LSTM は以下のように実装できる。まず、それぞれの LSTM のパラメータは共有すると考える。1 層目の RNN はシグナルの入力  $x_0, \dots, x_L$  を受け取った場合に、これは  $h_{L-1}$  を  $h_L$  と変換

\*2 なお組成モデルと同様な学習方式として、独立成分分析 (ICA) がある。独立成分分析も上記の Lloyd et al. のコンテキストと同様、回帰モデルと同様に時系列により統計的な規則を学習するため、われわれのモデルとは異なる。また、マルチタスク学習とも類似点は存在する。マルチタスク学習は、同じトレーニング集合を用いて異なるタスクに対して適用する。われわれの状況はマルチタスク学習と類似するが、マルチタスク学習において、各々のタスクに依存関係は存在しない。われわれのタスクは、一方のタスクで正解がわかれば、特徴量を差し引く/割り引くなどして、他方のタスクを解く際に影響を与える。したがって、依存マルチタスク学習だと言える。

し、さらにこの  $h_L$  を softmax により分類する ( $y$  と書き、便宜上、上向きに書くとする)。これだけであれば分類結果しかわからず、コンポーネントが 1 つの場合に  $f(x) (= x)$  を示すことしかしない。そこで内部関数として、クラス  $c$  ごとに  $f(x|c) : h_i, x_i, c \rightarrow x$  も同時に学習する。これにより、回帰と同様に対応する値を出力する。

トレーニング集合を用いて第 1 層目のみをもつ状態 (2 層目以上がない状態) で学習を行なうと、純粋に単独のコンポーネントを学習する区間、いくつかのコンポーネントの混合したセンサ出力に対して出力する区間が混じった形で出力される。いずれも、クラスの混合状態、出力の  $f(x)$  とその逆関数  $f^{-1}(y)$  を学習する。クラス数を  $C$  とすると、2 コンポーネントの混合に対しては  $C \times C$  の行列を用いて、すでにトレーニング集合で既出か否かをチェックし、出現していない組み合わせに関しては、テストフェーズのために組み合わせを想定しておく。2 つのコンポーネントを  $f_1(x), f_2(x)$  とし、これらの混合はスーパーポジション  $f(x) = f_1(x) + f_2(x)$  と考える。なお、2 つのコンポーネント間に依存関係があれば、これより小さい分布となるであろうが、今回の実験では省いた。以上が第 1 層目までの LSTM をもつ状態である。同様に、2 コンポーネントの混合に対しては  $C \times C \times C$  のテンソルを用いて、同様の準備を行なう。

テストフェーズにおいては 3 層の LSTM を用意する。単独のコンポーネントのみ、もしくは、トレーニング集合においても出現済みのコンポーネントの組み合わせに起因するセンサ出力であれば、トレーニング集合に基づいて学習した結果を用い、トレーニング集合において出現していないコンポーネントの組み合わせに起因するセンサ出力に対しては、上記で仮想する組み合わせに対して予測した結果を用いて推定を行なう。つまり、テストフェーズにおいては、トレーニングフェーズで用意した組み合わせに対するテストを行なう所が本論文の他の CNN+RNN モデルとは異なる部分である。

なお、本論文では前述したように認識可能なコンポーネント数の上限を固定している。つまり、3 層として、1 コンポーネント以上、3 コンポーネント以下の混合コンポーネントを同時に認識することを可能としているが、これは 4 コンポーネントの混合センサ出力が来ると対応できない。ただ、実験的な方法ということで留めておく。

なお、LSTM の出現前には「埋込みレーバ文法問題 (embedded Reber grammar 問題)」[2] という問題があったが、LSTM の出現により各々の入力値の記憶に対する問題を取り去ることにより解決されたので、ここにおいてその問題はないとする。

## 2.2 セミマルコフモデル

セミマルコフ連鎖モデルは隠れマルコフモデルを、各状

態  $s_i$  においてユニット長さでない時間  $d_i$  だけ留まってもよいと拡張する [9]. 時間  $d_i$  に達すると, システムは新たな状態  $s'_i$  に遷移して, これは  $s_i$  のみに依存する. このモデルにおいてシステムの振舞いは,  $i$  と  $i + d_i$  時間の間の時間においてマルコフ的ではなくなる. たとえば, 固有表現認識器 (NER) はエンティティのレベルの特徴を構築することによりより自然な形でイベントを捉えることができるようになる. 「時系列データのブロック」と「動作なし」の間では前ステップの動作に依存するのではなく, 忘れる操作を入れたい. この動作は CRF と semi-CRF の違いでもあり, semi-CRF においては, シークエンスの同定を終えると, 終了状態に入り, スタート状態が来たかどうかの最初のトリガーを探索する状態遷移を行なう.

LSTM には一見, 「忘却する」に関する語が多く存在するが, これは誤差逆伝播の勾配消滅 (もしくはバックプロパゲーション) に関する機構でありわれわれの欲しいものではない. たとえば LSTM の忘却ゲート (forget gate) は, 誤差信号を受取ればメモリセルで記憶した内容を忘れることを学習し, このことにより, これまでの記憶が不必要になった時点それぞれでセルを初期化することを可能とする. そうではなく, われわれの行ないたい忘却動作はフォワードプロパゲーションにおける「忘却」である.

GRU[1] をベースとして, セミマルコフゲートというものを導入することにより, これを実現することを考えたい. そのため, GRU を復習しておきたい. GRU は 2 つのゲートをもち, それぞれリセットゲート, 更新ゲートという ( $r_j$  をリセットゲート,  $z_j$  を更新ゲートとする). すると, GRU においては,  $j$  番目の  $h_j$  に対して以下の計算を行なう.

$$\begin{aligned} r_j &= \sigma([W_r x]_j + [U_r h_{<t-1>}]_j) \\ z_j &= \sigma([W_z x]_j + [U_z h_{<t-1>}]_j) \\ h_j^{<t>} &= z_j h_j^{<t-1>} + (1 - z_j) \tilde{h}_j^{<t>} \\ \tilde{h}_j^{<t>} &= \phi([W x]_j + [U(r \odot h_{<t-1>})]_j) \end{aligned} \quad (1)$$

なお  $\odot$  はアダマール積 (要素ごとの積) とする.

さて, これに最も単純な版として, 入力ゼロの場合にこれはセミマルコフ状態を示すというヒューリスティックを用いて単純化する版を考える. この場合, 前ステップの入力信号の状態により状態遷移を進行させるか留めるかの判断ができればよいので, セミマルコフゲートと呼ぶゲート一つを追加する. ゲート一つ追加しても, この GRU はまだ微分可能であるので誤差逆伝搬が可能となり, 通常通りの GRU の代替として用いることができる.  $m_j$  を GRU のセミマルコフゲートとし, 以下のように定義する.

$$\begin{aligned} r_j &= \sigma([W_r x]_j + [U_r h_{<t-1>}]_j) \\ z_j &= \sigma([W_z x]_j + [U_z h_{<t-1>}]_j) \\ m_j &= \sigma([W_m x]_j + [U_m h_{<t-1>}]_j) \\ h_j^{<t>} &= z_j h_j^{<t-1>} + (1 - z_j) \tilde{h}_j^{<t>} \end{aligned} \quad (2)$$

$$\tilde{h}_j^{<t>} = \phi([W x]_j + [U(r \odot m \odot h_{<t-1>})]_j)$$

## 2.3 畳み込み層

上述したように, 以上の 2 つの機構を本論文において導入する話題とし, CNN に関しては既出の 3 つのアーキテクチャをベースにすると述べた. これらに共通するのは, IoT の場合, センサ出力は 1D であるため, 畳み込み操作として 1D カーネルを用い, センサ出力の各次元に対して 1D カーネルを用いた畳み込み操作を独立に適用することである. また, プーリング層は用いていない. このような機構の下, CNN はフィルタ (もしくはカーネル) を用いて, 同じセンサ次元において時刻の異なる 2 点に対して畳み込みを行なう.  $(\ell - 1)$  層が入力を示すとすると, このとき  $(\ell)$  層は,  $K_{11}^{(\ell-1)}$ ,  $K_{21}^{(\ell-1)}$  という 2 つのカーネルを用いて抽出された 2 つの特徴写像  $a_1^\ell(\tau)$  と  $a_2^\ell(\tau)$  から構成される. さらに  $(\ell + 1)$  層は,  $(\ell)$  層の 1 次元のカーネル  $K_1^\ell$  による時間的な畳み込みから得られる単一の特徴写像からなる.

この機構において, CNN はセンサフュージョンと呼ばれる古典的な問題を解決する. センサフュージョンとは周波数の異なる 2 つの同種のセンサを同じ周波数のセンサのように取り扱う方法である. サンプル周波数がトレーニングフェーズとテストフェーズでは同じでない場合に対処でき, また RNN で処理するサンプリングの長さを短縮する場合にも対処できる.

## 3. 実験結果

### 3.1 データ

人工データ\*3は以下のように大きく 2 セット用意し, 1 セットは 3 種類のものを用意した. イベントなしが 6-7 割を占め, イベントがある場合は単一のラベルというデータセットを A, B, C という 3 種類用意した. 一方, イベントなしの区間はなく, イベントが単一の区間が 86% から 89% を占め, 2 イベントが同時に発生する区間がその他を占めるデータセットを D, E, F と 3 種類用意した. 詳細は表 2 に記した.

実装は Theano/Lasagne で行ない, 細部の構造は [7] と同

\*3 OPPORTUNITY データセット [8] を用いる予定であった. しかし, このデータセットはセンサ出力に対して, 行動が重なるような場面はあるが, ラベルのつけ方が常に単一で, 複数のラベルをつけている場面が存在しない. 本論文でのアルゴリズムはトレーニング集合において複数のラベルが存在し, その複数のラベルにより, 学習が進む部分が多い. もっとも, 単一のラベルに対し, 複数のラベルを推定する方法も許容する. しかし, 全体のアルゴリズムの性能を見たい部分が多く, したがって, このデータセットでの実験は諦め, 人工データに留まることに決めた. なお, OPPORTUNITY データセットは gestures と locomotion の 2 つのタスクからなり, gesture タスクでのセンサ出力は 113 次元をもち, 17 の動作を分類する. なお, われわれのアルゴリズムは, gesture タスクに対して, 単一のラベルにおいて, チューニングなしで以下のような性能をあげた.

(Yao et al., 2016b)	89.9
(Ordóñez et al., deepConvLSTM, 2016)	91.5
我々の実装	89.4

	A	B	C	D	E	F
イベントなし	763	745	769	0	0	0
単一ラベル	210	228	205	893	865	878
オーバーラップ (2つ)	0	0	0	107	135	122

表 2 2 グループのうち、最初のグループは A, B, C というデータセットであり、これはイベントなしが 6-7 割を占め、イベントがある場合は単一のラベルという内容をもつ。2 つ目のグループは D, E, F というデータセットであり、これはイベントなしの区間はなく、イベントが単一の区間が 86% から 89% を占め、2 イベントが同時に発生する区間がその他を占める。

様にし、畳み込みレイヤを 4 層もち、LSTM を 2 層、そして softmax の出力レイヤという構成である。窓のサイズを 24、窓のステップを 12、畳み込み操作の最終幅を 8、フィルタの数を 64、フィルタのサイズを 5、LSTM の隠れ層の数を 128、バッチサイズを 100、エポックを 500 とした。LSTM レイヤにはドロップアウト 0.3 を用いた。実験は Intel i7-7920HQ 3.10GHz 上で、NVIDIA GTX1080 上で cuDNN を用いて行なった。

### 3.2 実験

実験結果を表 3 に示した。A, B, C のデータセットにおいては単一のラベルしか存在しないので、組成インスタンスのような機能は必要ないため、ベースラインと組成インスタンスモデルの精度は同じとなった。セミマルコフモデルの精度は見込んだものとはならず、逆に若干落ちた。ヒューリスティックが若干弱かった可能性がある。

一方、D, E, F のデータセットにおいてはラベルの重複が存在し、ベースラインの精度が極端に悪い。このとき、組成インスタンスモデルを用いると、27% から 43% 上昇させることができた。しかし、全体としての精度は 20% 程度であり、若干満足な結果とは言えない。一方、セミマルコフモデルの精度は 1 つのデータセットに対して 1% 上昇させることができた。

	A	B	C	D	E	F
ベースライン	68.2	66.1	70.5	14.0	14.6	14.4
組成	68.2	66.1	70.5	20.0	19.6	19.4
セミ	68.0	65.9	70.0	13.8	14.7	14.4

表 3 ベースラインは CNN+RNN だが、組成インスタンスモデルもセミマルコフモデルもないものを指す。組成は、組成インスタンスモデルを用いた結果、セミはセミマルコフモデルを用いた結果を示す。いずれにおいても組成インスタンスモデルが他より良い結果を示す。

## 4. 結論

本論文においては、組成セミマルコフ畳み込み LSTM モデルを導入した。組成セミマルコフ畳み込み LSTM モデルは、セミマルコフモデル、組成インスタンスモデルを導入

し、深層センサモデルや深層 Deep 畳み込みモデルなどと同様の畳み込みモデルを考慮した。畳み込みモデルは次元毎に独立なモデルを導入する。実験結果が示すように、組成インスタンスモデルを用いることにより、27% から 43% に精度を向上させることができた。一方、セミマルコフモデルは単純なヒューリスティックをベースとしたため、あまり効果があったとは言えない結果となった。

さらなる挑戦は実際のデータセットで実験を行なうことである。脚注 3 にも述べたが多くのデータセットは、従来のトレーニング方法に最適なように単一のラベルをつけることが多い。このラベルのつけ方を自動で修正を加える方法などを考案したい。

### 参考文献

- [1] Kyunghyun Cho, Bart van Merriënboer Çağlar Gülçehre, Fethi Bougares, Holger Schwenk, Yoshua Bengio, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, EMNLP 2014.
- [2] Sepp Hochreiter and Jürgen Schmidhuber, “Long Short-Term Memory”, Neural Computation, 9(8):1735-1780, 1997.
- [3] Sozo Inoue, Naonori Ueda, Yasunobu Nohara, and Naoki Nakashima, “Mobile Activity Recognition for a Whole Day: Recognizing Real Nursing Activities with Big Dataset”, ACM Int’l Conf. Pervasive and Ubiquitous Computing (UbiComp), page 1269-1280, 2015.
- [4] Masaya Inoue, Sozo Inoue, Takeshi Nishida. “Deep Recurrent Neural Network for Mobile Human Activity Recognition with High Throughput”. arXiv:abs/1611.03607, 2016.
- [5] James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, Zoubin Ghahramani, “Automatic Construction and Natural-Language Description of Non-parametric Regression Models”, AAAI, page. 1242-1250, 2014.
- [6] 松木 萌, 井上 創造. “照度と電力消費量データおよび単語ベクトルを用いた行動推定の試み”, DICO, page. 1685-1690, 2016.
- [7] Francisco Javier Ordonez, Daniel Roggen. “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. Sensors 16:115, 2016.
- [8] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczeck, Gerhard Trster, Paul Lukowicz, Gerald Pirkl, David Bannach, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, and Jos del R. Milln. “Collecting complex activity data sets in highly rich networked sensor environments” In Seventh International Conference on Networked Sensing Systems (INSS’10), Kassel, Germany, 6 2010.
- [9] Sunita Sarawagi, William W. Cohen, “Semi-Markov COnditional Random Fields for Information Extraction”, 2004.
- [10] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor S Prentow, Mikkel B Kjrgaard, Anind Dey, Tobias Sonne, and Mads M Jensen, “Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition”, 13th ACM Confer-

ence on Embedded Networked Sensor Systems, 2015.

- [11] La The Vinh, Sungyoung Lee, Hung Xuan Le, Hung Quoc Ngo, Hyoung Il Kim, Manhyung Han, and Young-Koo Lee, "Semi-Markov conditional random fields for accelerometer-based activity recognition", Springer, 2011.
- [12] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, Tarek Abdelzaher. "DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing". arXiv:1611.01942v1, 2016.
- [13] Rui Yao, Guosheng Lin, Qinfeng Shi, Damith Ranasinghe, "Efficient Dense Labeling of Human Activity Sequences from Wearables using Fully Convolutional Networks", arXiv:1702.06212, page 1-7, 2017.