Deep Knowledge Tracingを用いた学習不振兆候の検出

令和2年1月26日

電気通信大学大学院 情報理工学研究科
情報・ネットワーク工学専攻 情報数理工学プログラム

学籍番号　1831074
史　博源
主任指導教員　植野　真臣教授
指導教員　川野　秀一准教授
hiromoto@ai.lab.uec.ac.jp

目 次

表 目 次

図 目 次

1　まえがき

学校でのデジタル学習環境の利用は，学習者の学習構成についてきめ細かいレベルで対応する新しい可能性をもたらしている．デジタル学習環境では，学習プロセスをよりよく理解するためのツールとデータを研究者に提供していると共に，学習者の学習経験を改善するための研究プラットフォームも与えている．

しかし，教師の指導や支援または，補完することを目的とした多くの学習支援システムでは，利用する学習者に限られた援助しか提供できない．その後の生産的な学習戦略を促進するため，より良い支援システムを開発し学習プロセスの中の学習者の躓きを検出することが重要である．難しい問題に遭遇したとき，その課題を解決するに十分な数の練習問題へ取りくみ続ける「固執」は，学習者にとって不可欠である．グリット [1]，忍耐力 [2]，及び生産失敗 (product failure)[3] などの研究から，持続性は学習者の成功に重要な役割を果たしていることが明らかになっている．学習プロセスにおいて，Wheel Spinning と Stopout は，持続性に関する学習不振兆候を示す重要な現象として考えられている [4]．

学習不振兆候の Stopout を示す学習者は，課題の早い段階で少数の問題を解くだけで学習をやめてしまい，追加の問題を通して難しい学習内容を学ぶ機会を逃している．一方，練習問題をやり続けるという固執の存在は，困難を乗り越える学習者にとって不可欠のものであるが，その固執が非生産的である場合もある．そのような非生産的な持続性という否定的な側面を示す兆候は，Wheel spinning として知られている [5]．具体的には，学習者が規定の学習課題に取り組みを続けていても制限時間内に学習の習熟状態に到達できない場合を指す．

Wheel spinning と Stopout はどちらも学習者の非生産的な学習兆候を表している．Stopout は成功するのに十分な持続性を示さない学習兆候を表し，Wheel spinning はインストラクターやチューターからの追加援助を求めることが学習者にとって有益である可能性が高い場合の学習兆候を表す．Beck と Cong の Wheel spinning に関する研究 [5] では，課題の

10 番目の問題までに課題を完了できない学習者が Wheel spinning として定義される．

学習者にとって有益な持続性を促進するために，Wheel spinning と Stopout を検出することが重要である．しかし，Stopout が検出されてから介入するとしても，学習者がすでにシステムの利用をやめている可能性が高い．同様に，Wheel spinning が検出されてから介入をするとしても，学習者がすでに時間と労力を浪費しており遅すぎる可能性がある．このような問題を避けるためには，学習者が Wheel spinning と Stopout を示す前に，そのような行動を見越して先行的に介入して，Wheel spinning と Stopout の潜在的原因に対処することが不可欠である．Botelho[4] らは，深層学習を用いて Wheel Spinnning や Stopout の予測を行ったが，学習者の知識状態が特徴量として用いられていないといった問題が考えられる．

一方で，知的学習支援システム (Intelligent Tutoring System:ITS) の分野において，学習データの分析を行うことで，学習過程における知識への習熟度や理解度を把握をすることが重要な研究テーマとなっている．学習者が課題の解決の知識をどのくらい獲得しているかという学習者の知識状態の推定を行うことにより，未習熟の課題を同定し，個人の成長に最適な指導を行うことを可能にする．これまで，学習過程における学習者の知識状態をモデル化し，過去の学習データから現在の知識状態を推定する手法が多く開発されている．一般的に知られているモデルとして Deep Knowledge Tracing:DKT がある [6]．DKT は深層学習を用いて過去の学習データと特徴量から学習者の知識状態を推定する．

また，Ritwick らの研究 [7] でヒント利用予測と知識状態推定を組み合わせた Multi Task Learning モデルが提案されており，結果としてヒント取得予測と知識状態推定の両方で精度が向上したことが報告されている．

これらを踏まえ，本研究の目的は，Botelho[4] らのモデルを拡張して，DKT による知識状態を特徴量として組み込むことで Wheel spinning と Stopout の早期検出を高精度に行うことである．具体的には，以下の手順

で行う．

1) 学習者の問題 ID とその問題の正誤から DKT を学習しその知識状態変数を取り出す．

2) 従来の学習不振兆候の特徴量と抽出した隠れ層を結合することで，新たに学習不振兆候予測の特徴量を生成する．

3) 新たに生成された特徴量を LSTM の入力として用いて学習不振兆候の予測を行う．

提案手法では．これまで学習不振兆候に用いられていない学習者の知識状態を特徴量として活用するだけでなく，知識状態の推定と学習不振兆候の予測を同時に行う Multi Task Learning を提案しており更に推定精度向上が期待できる．

Assistment2016-2017[4] を用いた評価実験において，提案手法は既存の LSTM モデルに比べ Wheel Spinning の分類精度が向上することが確かめられた．Stopout についても従来手法では推定が困難であった Stopout の検出が可能となったことが確かめられた．

本論文は，以下の構成からなる．まず先行研究について第 2 章にまとめる．その後，第 3 章では提案手法の概要を示し，第 4 章では今回の評価実験に用いるデータセットについて説明したあと，比較実験によりその効果を明らかにする．最後に，第 5 章において本論文のまとめと今後の課題を示す．

2　先行研究

近年，Congnitive tutor[8] や ASSISTments[5],[9] や大規模オープンオンラインコース MOOC[10] などのオンライン学習プラットフォームが提供され，その上で Wheel spinning のモデル化など多くの学習の持続性に関する研究成果が報告されている [5][8][9][10].

2.1　Wheel spinning

Wheel spinning は学習者が学習課題に解き続けるが，学習教材の十分な理解を得ることができない行動を指す． Wheel spinning の用語は，雪や泥の中で立ち往生している車が必死に動いているにもかかわらず車輪が空転している状態に由来している．Beck と Cong の Wheel spinning に関する研究 [5] では，Wheel spinning を 10 回チャレンジしたにも関わらず要求された習熟度に到達できない状態であると定義している．

課題にある問題をすべて答えることを学習者に要求する伝統的なの課題の出題方法に対して，ASSISTments など習熟度ベースに基づく課題の出題方法では，課題を完了するために割り当てられたスキルに対する十分な理解，または習熟を示すことを学習者に要求する．例えば，ASSISTments の場合，学習者がすべての問題を解くのではなく，ヒントを利用せずに 3 つの問題に連続して正解することを合格と見なしている．

Wheel spinning を観測する初期の研究は，習熟度ベースの課題との学習者の相互作用に着目したものである [9]．このようなモデルでは，各問題と学習者の直近の行動について，専門家が作成した特徴量を用い，現在の課題で学習者が Wheel spinning である可能性を推定していた．

また，Botelho[4] らの Wheel spinning に関する研究では，より細かい粒度の説明変数を導入し，深層学習の手法を用いて Wheel spinning の動作を長期間にわたって予測した．さらにモデルのパフォーマンスがどのように連続した問題で変化するかを考察している．

2.2 Stopout

最初の少数の問題で学習をやめてしまう現象を，Stopout という．学習者が Stopout することでより困難な教材を学ぶ機会を逃してしまう．

学習者が授業や課題の受講をやめてしまうことについては，より一般的には学習者の Dropout として，主に MOOC などのデジタル学習環境における教育の問題として大きな注目を集めている [10],[11],[12],[13] ．

Dropout は，学習者によって理由が異なることが先行研究 [14] によって示されている．例えば，不十分な背景知識やコンテンツの難しさのためにやめてしまう人もいれば，時間管理やスケジュールのせいでやめてしまう人もいる．MOOC 内の学生の Dropout は,「Gritnet」という学習モデルの開発を通じて以前から研究されている [14].

Dropout と Stopout の違いは，学習者は Stopout の状態であっても講義を受講しており，その後の課題を完了することを選択できることである．一方で講義の受講自体をやめてしまう場合は，Dropout として定義される．学習者の Dropout が，学習者が教材を十分に習得することを妨げ，その後必要なスキルを習得する際にさらなる困難につながる可能性がある [15]．Stopout を示す学習者は，学習環境との相互作用をやめてしまっているため，プラットフォームを介して学習者を支援することができず，その場合学習者を助けるために教師などの外部の力に頼ることしかできない．そのため，すでに Stopout を示している学習者の Stopout の原因を特定し，生産性の高い持続性をサポートするための効果的な介入をすることは，困難である．

これらの理由により，課題内での学習不振兆候を防ぐため，Stopout を示す可能性のある学習者をなるべく早い段階で判断することができるモデルを構築することが重要である．

2.3 教育分野における深層学習

教育や学習分析などの分野では，学習者の行動や成績の予測について，深層学習を利用した研究が増えている．特に，教育分野では，学習者の行動の複雑な時系列パターンをモデル化する能力が高いため，多くの研究で再帰型ニューラルネットワーク（recurrent neural network：RNN)[16] が使われている．特に RNN の一種である Long-Short Term Memory(LSTM)[17] は，画像処理や自然言語処理など広範囲の分野に用いられてきた．Botelho ら [4] の Wheel Spinning，Stopout の検出にも LSTM が用いられている．その他にも LSTM を用いて，学習者の知識と短期間の成績予測 [6][18][19]，MOOC における学習者の卒業予測 [14] とリアルタイム成績 [20]，学習者の感情状態の検出 [21]，長期的な成績予測 [22][23] などが行われている．

深層学習モデルは，高精度に予測が可能であることが報告されているにもかかわらず，多数の学習パラメータと複雑なモデル構造はそれらの解釈を困難にしてしまう．しかし，パラメータの解釈性は存在しないとしても，深層学習の隠れ層にはそれまでの豊富な情報が含まれているとみなせ，別のモデルの特徴量として用いることは可能である．

2.4 Deep Knowledge Tracing(DKT)

本節では，知識状態推定モデルについて先行モデルの説明を行う．本研究では学習履歴データにおける問題数を M と表し， 学習者の問題 m に対する反応データ x_m を次のように表す.

$$x_m = \begin{cases} 1 & (\text{学習者が問題 } m \text{ に正答}) \\ 0 & (\text{上記以外}) \end{cases} \tag{1}$$

Deep Knowledge Tracing(DKT) は，それまでの研究で考えられていたスキル間の独立やマルコフ過程を仮定せず，過去の学習データから時系列の深層学習モデルである LSTM を用いて課題への反応を予測するモデルである [6].

まずはじめに，LSTM のモデル構造についてである．LSTM は，RNN の学習時に入力長が長くなると指数関数的に勾配が小さくなってしまう勾配消失問題 [24] を解決するために長期依存を保存しているセルとゲートベクトルが導入されている．時点 t における LSTM のモデルを図 1 に示す．具体的には，X の時点 t における入力 $\boldsymbol{X}_t$ と $t-1$ の隠れ層 $\boldsymbol{h}_{t-1}$ をを用い式 (2)(3) で忘却ゲート $\boldsymbol{f}_t$，入力ゲート $\boldsymbol{i}_t$ の値を求め式 (6) を用いて m_t を更新する．忘却ゲート $\boldsymbol{f}_t$ は，セルが保存している長期依存をどのくらい維持するかを調節し，入力ゲート $\boldsymbol{i}_t$ は，新たにセルに保存する値を調整する役割を果たす．これらのゲートベクトルの導入により $\boldsymbol{m}_t$ に保存する情報と削除する情報を制御することが可能となっており，長期依存を扱うことができるようになっている．さらに式 (4) によって求められる出力ゲートとセルを用いて，式 (7) のように時点 t における出力となる隠れ層 $\boldsymbol{h}_t$ を求める．

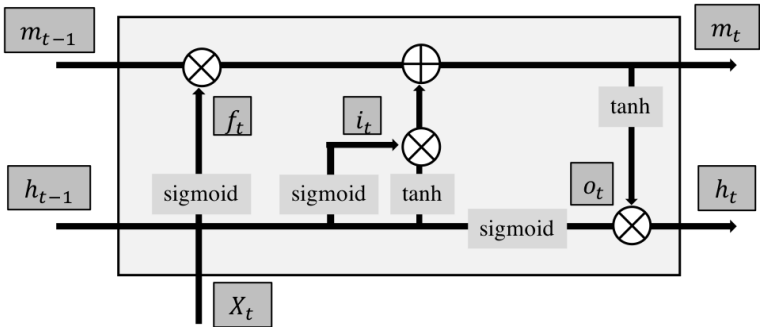

図 1: LSTM のモデル構造

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{fx}\boldsymbol{X}_t + \boldsymbol{W}_{fh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \tag{2}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{ix}\boldsymbol{X}_t + \boldsymbol{W}_{ih}\boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \tag{3}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{ox}\boldsymbol{X}_t + \boldsymbol{W}_{oh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \tag{4}$$

$$\boldsymbol{g}_t = \sigma(\boldsymbol{W}_{gx}\boldsymbol{X}_t + \boldsymbol{W}_{gh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_g) \tag{5}$$

$$\boldsymbol{m}_t = \boldsymbol{f}_t \circ \boldsymbol{m}_{t-1} + \boldsymbol{i}_t \circ \tanh(\boldsymbol{g}_t) \tag{6}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh(\boldsymbol{m}_t) \tag{7}$$

$\boldsymbol{W}_{fx}$，$\boldsymbol{W}_{fh}$，$\boldsymbol{W}_{ix}$，$\boldsymbol{W}_{ih}$，$\boldsymbol{W}_{ox}$，$\boldsymbol{W}_{oh}$，$\boldsymbol{W}_{gx}$，$\boldsymbol{W}_{gh}$ は重みベクトルであり，$\boldsymbol{b}_f$，$\boldsymbol{b}_i$，$\boldsymbol{b}_o$，$\boldsymbol{b}_g$ はバイアスベクトルである．これらの次元数は，任意に設定する隠れ層の次元数と等しく，全て同時に学習される．σ は式 (8) に示すシグモイド関数であり，$tanh$ は式 (9) に示すハイパボリックタンジェントである．ここで $\circ$ はアダマール積を表している．

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{8}$$

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{9}$$

以下では式 (2) から式 (7) をまとめて次式で表す．

$$\boldsymbol{h}_t = LSTM(\boldsymbol{X}_t, \boldsymbol{h}_{t-1}) \tag{10}$$

次に DKT のモデル構造についてである．DKT では時刻 t までの学習者の課題への反応ベクトル Q_t^{DKT} を入力データとし，時刻 t における各スキルに対する予測反応ベクトル $\boldsymbol{y}_t^{DKT}$ を出力する． 時刻 t の LSTM の隠れ層を h_t^{DKT} として表すとき DKT モデルは以下の図 2 で表される

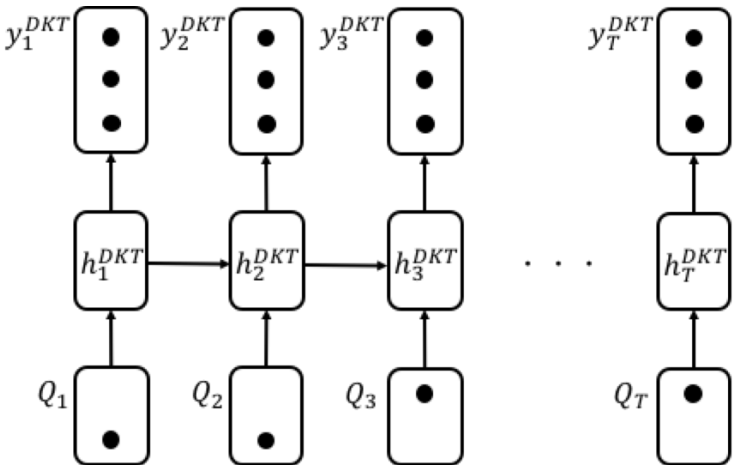

図 2: DKT のモデル構造

DKT の入力は，問題数 M のデータセットについて，固定長ベクトル P_t をもとにワンホットベクトル Q_t を作成してモデルの入力として用いる．

$$\boldsymbol{P}_t = \{\boldsymbol{q}_t, a_t\} \tag{11}$$

$$\boldsymbol{Q}_t \in \{0,1\}^{2M} \tag{12}$$

$$\boldsymbol{Q}_t = \begin{cases} [\boldsymbol{0}_M, \boldsymbol{q}_t] (a_t = 0 \text{ の時}) \\ [\boldsymbol{q}_t, \boldsymbol{0}_M] (\text{上記以外}) \end{cases} \tag{13}$$

ここで q_t は時点 t で解いた問題 ID の要素を 1 とするワンホットベクトルであり，a_t は時点 t で解いた問題の正誤を表す．また，$\boldsymbol{0}_M$ は長さ M のゼロベクトルを意味する．

$\boldsymbol{P}_t, \boldsymbol{q}_t, a_t, \boldsymbol{Q}_t$ について例を用いて説明する．

ex : 問題 $id1$ に正解した場合, 問題数 3

$$\boldsymbol{p}_t = \{(1,0,0), 1\} (M = 3) \rightarrow \boldsymbol{Q}_t = \{1,0,0,0,0,0\}$$

ex : 問題 $id2$ に誤答した場合, 問題数 3

$$\boldsymbol{p}_t = \{(0,1,0), 0\} (M = 3) \rightarrow \boldsymbol{Q}_t = \{0,0,0,0,1,0\}$$

DKT の隠れ層 $\boldsymbol{h}_t^{DKT}$ と出力 $\boldsymbol{y}_t^{DKT}$ は式 (10) を用いて次のように表される．

$$\boldsymbol{h}_t^{DKT} = LSTM(\boldsymbol{Q}_t, \boldsymbol{h}_{t-1}^{DKT}) \tag{14}$$

$$\boldsymbol{y}_t^{DKT} = \boldsymbol{W}_{h_t^{DKT}} \boldsymbol{h}_t^{DKT} + \boldsymbol{b}_{h_t}^{DKT} \tag{15}$$

このとき $\boldsymbol{W}_{h_t^{DKT}}$ は重みベクトルを表し，$\boldsymbol{b}_{h_t}$ はバイアスベクトルを表す．DKT のパラメータ学習は，次式ように，次の時点 $t+1$ で解く項目に対する反応予測の誤差を示す損失関数を最小化することで実現できる．

$$loss_{DKT} = \sum_{t} l((\boldsymbol{y}^{DKT})^{\top} \boldsymbol{q}_{t+1}, a_{t+1}) \tag{16}$$

ここで q_{t+1} は時点 $t+1$ で実際に学習者が解答した問題ベクトルを表し，a_{t+1} は時点 $t+1$ での問題への正誤を表す．l は交差エントロピー誤差関数であり式 (17) のように表される．

$$l(q_t, a_t) = -q_t \log a_t - (1 - q_t) \log(1 - a_t) \tag{17}$$

3　提案手法

3.1　提案手法の流れ

本研究では，DKT を用いて知識状態を考慮した新たな学習不振兆候予測の手法を提案する．学習不振兆候予測とは，Wheel Spinning,Stopout それぞれの検出のことを指し，提案手法のモデルは共通である．時点 T までの提案手法の概要を図 3 に示す．提案手法の流れは以下の通りである．

1) 学習者の問題 ID とその問題の正誤から DKT を学習し，その隠れ層を抽出することで，知識状態変数を取り出す．

2) 従来の学習不振兆候のデータセットに含まれる特徴量と抽出した隠れ層を結合することで，学習不振兆候予測の特徴量を作成する．

3) 新たな特徴量を LSTM の入力として学習不振兆候の予測を行う．

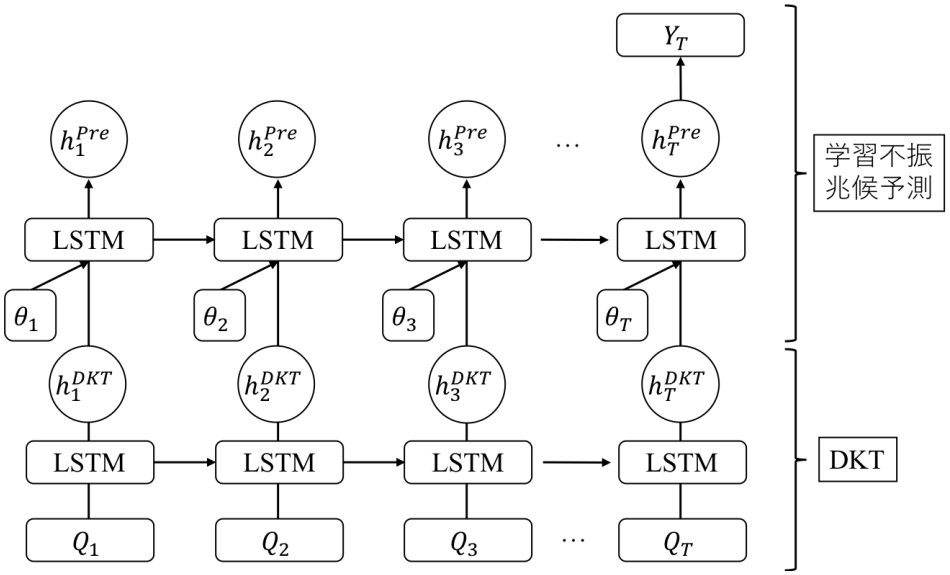

図 3: 提案手法の概要

提案手法の実装では，LSTM の実装にニューラルネットワークのフレームワークの一つである $Chainer$[2] を用いた．L ミニバッチ数を 100，エポック数を 30 に固定した．またパラメータの最適化アルゴリズムには　adaptive moment estimation(Adam)[25] を用いた．

まずはじめに，図 3 中の DKT の部分について説明する．DKT では，解いた問題とその正誤を one-hot ベクトルで表す．問題数が M のデータセットについて p_t を以下のように定義する．

$$\boldsymbol{p}_t = \{\boldsymbol{q}_t, a_t\} \tag{18}$$

ここで $\boldsymbol{q}_t$ は時点 t で解いた問題 ID の要素だけが 1 となるような one-hot ベクトルであり，a_t は時点 t で解いた問題の正誤を表す．a_t は次のように定義する．

$$a_t = \begin{cases} 1 & (\text{学習者が時点 } t \text{ で解いた問題に正答}) \\ 0 & (\text{上記以外}) \end{cases} \tag{19}$$

次に $\boldsymbol{p}_t$ を固定長のワンホットベクトル $\boldsymbol{Q}_t$ に変換することで DKT の入力として用いる．

$$\boldsymbol{Q}_t \in \{0, 1\}^{2M} \tag{20}$$

$$\boldsymbol{Q}_t = \begin{cases} [\boldsymbol{0}_M, \boldsymbol{q}_t] (a_t = 1 \text{ の時}) \\ [\boldsymbol{q}_t, \boldsymbol{0}_M] (\text{上記以外}) \end{cases} \tag{21}$$

ここで $\boldsymbol{0}_M$ は M 次元の 0 ベクトルである．次に，2.3 節で示したように式 (10) を用いて，DKT の隠れ層の値 $\boldsymbol{h}_t^{DKT}$ を次式のように定義する．

$$LSTM(\boldsymbol{Q}_t, \boldsymbol{h}_{t-1}^{DKT}) = \boldsymbol{h}_t^{DKT} \tag{22}$$

このとき出力 y_t^{DKT} は次式で表される．

$$\boldsymbol{y}_t^{DKT} = \boldsymbol{W}_{h_t^{DKT}} \boldsymbol{h}_t^{DKT} + \boldsymbol{b}_{h_t} \tag{23}$$

本研究では，求めた隠れ層 $\boldsymbol{h}_t^{DKT}$ の値を学習不振兆候予測の特徴量として用いる．学習不振兆候予測において 3.2 節で示した従来の特徴量を θ_t とする. この特徴量 θ_t と $\boldsymbol{h}_t^{DKT}$ を結合することで，新たな学習不振兆候の特徴量 $\boldsymbol{x}_t^{Pre}$ として用いる．この時，$\boldsymbol{x}_t^{Pre}$ は次式で表される．

$$x_t^{Pre} = [\theta_t; h_t^{DKT}] \tag{24}$$

; は，θ_t と $\boldsymbol{h}_t^{DKT}$ のベクトルの結合を表す．

DKT と同様に学習不振兆候の隠れ層 $\boldsymbol{h}_t^{Pre}$ は式 (25) で表される．また学習不振兆候予測においては LSTM の最後入力時点 T の隠れ層のみを用いる．この最後の隠れ層 $\boldsymbol{H}_{last}^{pre}$ を式 (26) に示す．

$$LSTM(\boldsymbol{x}_t^{Pre}, \boldsymbol{h}_{t-1}^{Pre}) = \boldsymbol{h}_t^{Pre} \tag{25}$$

$$\boldsymbol{H}_{last}^{pre} = \boldsymbol{h}_T^{Pre} \tag{26}$$

学習不振兆候予測の出力層 $\boldsymbol{y}_t^{Pre} = [\boldsymbol{y}_t^{Pre=0}, \boldsymbol{y}_t^{Pre=1}]$ と出力 $\boldsymbol{Y}_t$ はそれぞれ次式で定義する．

$$y_t^{Pre} = \boldsymbol{W}_y \boldsymbol{H}_{last}^{pre} + \boldsymbol{b}_y \tag{27}$$

$$Y_t = S(\boldsymbol{y}_t^{Pre}) \tag{28}$$

ここで，$\boldsymbol{W}_y$ は重みベクトルを表し，$\boldsymbol{b}_y$ は，バイアスベクトルである．また $S(\boldsymbol{y}_t)$ はソフトマックス関数を表し, 次式で表される ．

$$S(\boldsymbol{y}_t^{Pre}) = \left[\frac{y_t^{Pre=0}}{y_t^{Pre=0} + y_t^{Pre=1}}, \frac{y_t^{Pre=1}}{y_t^{Pre=0} + y_t^{Pre=1}} \right] \tag{29}$$

また出力 $\boldsymbol{Y}_t$ は，次式のような 2 値を持つ．

$$\boldsymbol{Y}_t = \begin{cases} 1 & (\text{学習者が} \mathit{WheelSpinning} \text{ または } \mathit{Stopout} \text{ している}) \\ 0 & (\text{上記以外}) \end{cases} \tag{30}$$

3.2 パラメータ学習

一般に，深層学習では微分可能な損失関数を定義し，誤差逆伝播法によりパラメータを学習を行う．損失関数を最小化することでパラメータ

学習を実現する．提案モデルの損失関数は，DKT の損失関数と学習不振兆候予測の損失関数の重み付き和で求める．

一般の機械学習手法と同様に，深層学習もデータの偏りに大きな影響を受けることが知られている．そこで，データの偏りの問題を解決するため出現頻度の少ないデータの重みを大きくする cost-sensitive learning が一般に用いられている [26]．本研究でも，交差エントロピー誤差関数の l を用いて DKT の損失関数と学習不振兆候予測の損失関数を次の式 (31) と式 (32) で定義する．

$$loss_{DKT} = \sum_{t} l((y^{DKT})^{\top} q_{t+1}, a_{t+1}^{DKT}) \tag{31}$$

$$loss_{Pre} = \sum_{t} l(y_t^{Pre}, y_t^{\hat{Pre}}) + \gamma \sum_{t \in y_t^{Pre}=1} l(y_t^{Pre}, y_t^{\hat{Pre}}) \tag{32}$$

a_{t+1}^{DKT} は時点 $t+1$ での問題への正誤を表す．$y_t^{\hat{Pre}}$ は実際の反応データを表す．式 (32) において γ の値は，通常時 0 であるが，少数ラベルの偏りを考慮して重み付けを行う場合 $\gamma = 1.0$ として少数ラベルに重みを付けて計算する場合がある．

提案モデルの損失関数は次式で表される．

$$loss = \alpha_1 \times \ loss_{DKT} + \alpha_2 \times \ loss_{Pre} \tag{33}$$

α_1 と α_2 はそれぞれのパラメータの尺度の違いを考慮して $\alpha_1 = 0.001$，$\alpha_2 = 1$ として損失関数に重み付けを行い，パラメータ推定を行った．

4　評価実験

4.1　データセット

評価実験で用いるデータセットについて説明する．

4.1.1　ASSISTments

一般に不振兆候予測タスクで使用されているデータセットとして ASSISTments データセットがある. このデータセットは，2016 − 2017 年度に ASSISTments に取り組んだ学習者の学習履歴から構成されている．ASSISTments は，Web ベースの学習プラットフォームであり，教師には授業や宿題を割り当てるためのツールが提供され，学習者は即座に正解のフィードバックを受けられる．それぞれの課題に取り組んでいる間，学習者に対して即座に援助を与えることが可能である．更に必要になる場合は，問題を解くための過程をより小さなステップに分割し，適切なヒントを学習者に提供する．

これらに加えて ASSISTments は，学習者が正しく答えられるまで次の問題に進むことを許さないため，学習者が問題を正しく答えることができない場合は，学習者に最終ヒント (bottom-out hint) を提供することも可能である．

ASSISTments 内の課題の完了を示す基準は，システムの援助（あらゆるヒント）を受けずに連続して 3 つの問題を正しく答えられることである．

学習者がスキルをマスターできない場合は，ASSISTments はその日の新たな課題の割り当てを停止し，追加のヒントを提供する．

ASSISTments は，平均 1 日で数千人の異なる学習者によって使用される．その殆どは主に数学の問題を解く 6 年生から 8 年生であり，深い学習支援を行うために十分な規模とバリエーションをもつデータセットが提供されている．データセットには，これらの学習履歴がすべて保存されているが，多くの研究で実際に用いられているのは少なくとも 10 名以上が解いている問題，かつ全体の課題を終わらせた割合が少なくとも 70

%である学習者の履歴のみを用いる．この制限で，補足課題などの実験に有効でないデータを取り除くことができる．

4.1.2 データセットの特徴

4.1.2.1 特徴量

データは，ASSISTments によって記録されている学習行動で構成され，きめ細かいレベルの情報が数値化されている．データの各行には，時間に関連した情報に加えて問題の正誤や要求したヒントなどの 15 個の特徴量から，one-hot-encoding が適用され，合計 71 個の特徴量が含まれる．

これらの特徴量についての説明を表 1 に示す．また特徴量のデータサンプルについてを表 2 に示す

表 1: 学習不振兆候予測の特徴量とその説明

特徴量の名前	特徴量の説明
Action Type	問題に解答，ヒントの要求などどのような行動をその問題に対してしたのか
Attempt Count	現在の行動まで問題に解答をした数
Hint Count	現在の行動でヒントを要求した数
Problem Count	現在の行動までに解いた問題の数
Probability of Action	与えられた問題に対して学習者の現在の行動の確率
Probability of Action Given Action Count	問題と行動の数が両方与えられたときの 学習者の現在の行動の確率
Probability of Response	現在の問題の解答の中で学習者が特定の解答をする確率
Probability of Response Given Action Count	問題と問題で実行された問題の数の両方を与えた状態での 現在の問題の解答の中で学習者が特定の解答をする確率
Cumulative Log Likelihood	問題に対する特定の解答で解答する学生の解答累積対数尤度
Normalized Time Taken	行動の種類と問題に対応した最後の行動からの経過時間
Used penultimate Hint	現在の問題の中で最後から2番目のヒントが使われるかどうか
Used Bottom Out Hint	現在の問題の中で最後のヒントが使われたかどうか
Correctness	現在の問題に対して，正解，不正解．解答しなかったの何れであるか
Preceding 3 Actions	現在の問題を含む直前3課題について学習者がどのような行動をしたのか
Current and Preceding 2 Actions	現在の課題を含む直前2課題について学習者がどのような行動をしたのか

本研究では，DKT の隠れ層ベクトルを提案しており，81 個の特徴量を提案モデルへの入力として用いた．

表 2: 学習不振兆候予測の特徴量とデータサンプル

特徴量	データサンプル
attempt_count	1
hint_count	0
problem_count	1
probability_action	0.8689035
probability_action_action_count	0.7674862
probability_answer	0.6923077
probability_answer_action_count	0.8
log_likelihood_cumulative_answer	-0.3677248
normalized_time	-0.02797081
used_penultimate_hint	0
used_bottom_out_hint	0
fold	0
action_name_answer	1
action_name_answerhint	0
action_name_hint	0
action_name_scaffold	0
correct_0.0	0
correct_1.0	1
correct_nan	0
previous_3_actions_answer_answer_answer	0
previous_3_actions_answer_answer_hint	0
previous_3_actions_answer_answer_scaffold	0
previous_3_actions_answer_hint_answer	0
previous_3_actions_answer_hint_answerhint	0
previous_3_actions_answer_hint_hint	0
previous_3_actions_answer_scaffold_answer	0
previous_3_actions_hint_answer_answer	0
previous_3_actions_hint_answer_hint	0
previous_3_actions_hint_answerhint_answer	0
previous_3_actions_hint_hint_answer	0
previous_3_actions_hint_hint_hint	0
previous_3_actions_null_answer_answer	0
previous_3_actions_null_answer_answerhint	0
previous_3_actions_null_answer_hint	0
previous_3_actions_null_answer_scaffold	0
previous_3_actions_null_hint_answer	0
previous_3_actions_null_hint_hint	0
previous_3_actions_null_null_answer	0
previous_3_actions_null_null_hint	0
previous_3_actions_null_null_null	1
previous_3_actions_null_null_scaffold	0
previous_3_actions_null_scaffold_answer	0
previous_3_actions_scaffold_answer_answer	0
current_and_past_2_actions_answer_answer_answer	0
current_and_past_2_actions_answer_answer_hint	0
current_and_past_2_actions_answer_answer_scaffold	0
current_and_past_2_actions_answer_answerhint_answer	0
current_and_past_2_actions_answer_hint_answer	0
current_and_past_2_actions_answer_hint_answerhint	0
current_and_past_2_actions_answer_hint_hint	0
current_and_past_2_actions_answer_scaffold_answer	0
current_and_past_2_actions_answer_scaffold_scaffold	0
current_and_past_2_actions_answerhint_answer_answer	0
current_and_past_2_actions_hint_answer_answer	0
current_and_past_2_actions_hint_answer_hint	0
current_and_past_2_actions_hint_answerhint_answer	0
current_and_past_2_actions_hint_hint_answer	0
current_and_past_2_actions_hint_hint_hint	0
current_and_past_2_actions_null_answer_answer	0
current_and_past_2_actions_null_answer_answerhint	0
current_and_past_2_actions_null_answer_hint	0
current_and_past_2_actions_null_answer_scaffold	0
current_and_past_2_actions_null_hint_answer	0
current_and_past_2_actions_null_hint_hint	0
current_and_past_2_actions_null_null_answer	1
current_and_past_2_actions_null_null_hint	0
current_and_past_2_actions_null_null_scaffold	0
current_and_past_2_actions_null_scaffold_answer	0
current_and_past_2_actions_null_scaffold_scaffold	0
current_and_past_2_actions_scaffold_answer_answer	0
current_and_past_2_actions_scaffold_answer_scaffold	0

表 3: データセットの各データ総数

学習者数	10613
課題総数	4384
問題総数	28223
総アクション(行動)数	616692

4.1.2.2 Wheel spinning と Stopout のラベル付けについて

Botelho[4] らは次のように Wheel spinning と Stopout のラベル付けを行っている．Wheel spinning と Stopout をそれぞれ排他的なものとして行っている．

まずはじめに，学習者が 10 回目の問題で十分な理解の閾値に達していない場合に Wheel Spinning とラベル付けしている．Wheel spinning を判断するための 10 個目の問題という設定は，今後の研究でより議論される必要があり，任意である．

Stopout は，10 番目の問題の前に，問題に取り組むことをやめた場合にラベル付けされる．

Wheel spinning と Stopout のラベルは，学習者の課題単位で計算される．提案モデルでは先行研究をもとに学習者の各行動から Wheel spinning と Stopout に陥るかを予測する．さらに，提案モデルでは時系列のデータを活用して Wheel spinning と Stopout を推定することが可能である．

また，本研究では学習者が 2 問以上解答した課題のデータを抽出し，表 3 で示す 10613 人の学習者，4384 個の課題，28223 個の問題，616692 行の行動データを含む実験用データセットを作成した．

4.2 評価実験の方法

提案手法が，ASSISTMENT データセットを用いて既存手法よりも提案手法のほうが高い分類精度であることを示す．

提案手法に加えLSTM[17]，Support Vector Machine (SVM) [27]，Naive Bayes (NB)[28]，Neural Network (NNET)[29]，K 近傍法 [30] ，Random Forest (RF)[31] を用いて 5 分割交差検証を行い，Accuracy(精度)，それぞれの値についての適合率 (precision) と再現率 (Recall) の調和平均である F 値とその平均を求めた．提案手法と LSTM モデルは，学習者の各課題毎の時系列を考慮して入力を行うが，その他の，SVM，NB，NNET，K 近傍法，RF については，学習者の各課題毎の時系列を考慮することができないため，各時点ごと独立に予測を行った．

Wheel Spinning と Stopout の検出に用いる入力については，知識状態変数を含まない場合は，データセットの特徴量を入力として用い，知識状態変数を含む場合は，問題 ID(problem id) とその問題の正誤の 2 値を入力として加えて評価を行った．

提案手法では，ミニバッチ処理を行い分類を行った．このときミニバッチ数は 100 で，試行回数は 30 回である．また，式 (32) において γ の値を $\gamma = 0$ と $\gamma = 1.0$ とし損失関数に対して，少数ラベルの重み付けの有無についても評価を行う．DKT の隠れ層のノード数と学習不振兆候の隠れ層のノード数は 10 に固定して推定を行う．

次に評価実験に使用した LSTM モデルについて示す．f_t は提案手法と同じ学習不振兆候の特徴量である．隠れ層 h_t^{Pre} のノード数は 10 である．エポック数は 1500 に固定して行った．

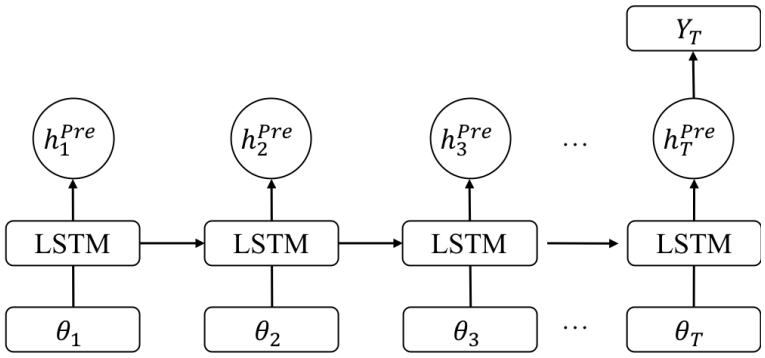

図 4: LSTM モデル

4.3 Wheel Spinning の評価実験

まずはじめに，Wheel Spinning の検出について提案モデルの損失関数は次の図 5 のようになった．図 5 より，エポック数が 3000 の時の損失関数が約 0.2 程度となっており，正しく学習されていることが確かめられた．

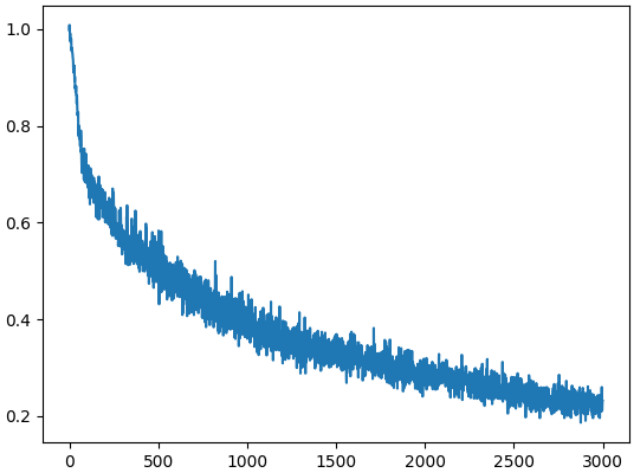

図 5: 提案手法 ($\gamma = 0$) の Wheel Spinning 検出の損失関数

次に Wheel Spinning について分類結果についてまとめる．Wheel Spinning の分類結果を表 4 に示す．

分類結果について，精度，各 F 値，F 値の平均に分けて考察を行う．

まずはじめに，精度についてである．従来の LSTM モデルが 0.924 であるのに対し，提案手法である知識状態変数を含む LSTM モデルが 0.941 となり約 2% 程度分類精度が向上したことが確認できた．少数ラベルの損失関数に重み付けを行う場合との比較では，少数ラベルに重み付けを行わないモデルの方が 5% 上回る結果となった．またその他の SVM などの従来手法と比較しても約 10% 程度上昇しており，どのモデルよりも提案手法が高精度に Wheel Spinning の検出が可能であることが確かめられた．したがって，知識状態変数を推定に加えて Multi Task Learning することにより知識状態変数を適切に活用できていることが確かめられた．

表 4: Wheel Spinning の分類結果

評価項目 / モデル	Accuracy	0のF値	1のF値	F値の平均
提案手法(γ = 0)	0.941	0.968	0.650	0.809
提案手法(γ=1.0)	0.891	0.936	0.635	0.785
LSTM	0.924	0.959	0.468	0.714
SVM	0.804	0.869	0.614	0.742
NB	0.336	0.034	0.494	0.264
NNET	0.805	0.865	0.651	0.758
K近傍法	0.806	0.867	0.640	0.754
RF	0.805	0.867	0.635	0.751

次に各 F 値についてである．ラベル 0 の F 値について，少数ラベルの損失関数に重み付けをしない提案手法と従来の LSTM モデルを比較すると，提案手法が 0.968 であるのに対し，従来の LSTM モデルは 0.959 であり，0 の F 値においても精度が向上したことが確かめることができた．少数ラベルの損失関数に重み付けを行う提案手法と比較した場合では，わずかに少数ラベルの損失関数に重み付けを行わないモデルの方が上回る結果となった．その他の SVM などの従来手法と比較した場合もすべての場合で提案手法が優れており，提案手法の有効性が確かめられた．

ラベル 1 の F 値について，少数ラベルの損失関数に重み付けを行わない提案手法よりも，NNET がわずかに上回っている．少数ラベルに重み付けを行うモデルよりも重み付けを行わないモデルのほうが 1 の F 値は高くなった．少数ラベルの損失関数に重み付けを行わない提案手法の 1 の F 値が 0.650 であるのに対して従来の LSTM モデルの 1 の F 値は 0.468 と提案手法が大きく上回っているので，知識状態変数を加えることで，少数ラベルの分類精度も上昇することが確かめられた．

最後にラベル 0 の F 値とラベル 1 の F 値を平均して求めた F 値の平均についてである．F 値を平均した場合は，提案手法が 0.809 であるのに対して，従来の LSTM モデルは 0.714 であり，約 10% 程度上昇することが確かめられた．さらに少数ラベルの損失関数に重み付けを行うモデルより，行わないモデルの方が F 値の平均も高い結果となった．その他の SVM などのモデルと比較しても約 5% 程度上昇しており，提案手法を用いることで全体的な分類精度を上昇することを確かめることができた．

以上の結果をまとめる．提案手法のうち少数ラベルの損失関数に重み付けを行うモデルと行わないモデルでの比較では，少数ラベルの損失関数に重み付けを行うことによる各精度の上昇は見られなかった．従来手法との比較では，提案手法のモデルの推定精度や F 値の値が上回っており，Wheel Spinning の検出において，提案モデルが従来手法よりも高精度で推定できることが明らかになった．

4.4 Stopout の評価実験

まずはじめに，Stopout の検出について提案モデルの損失関数は次の図 6 のようになった．図 6 より，エポック数が 3000 の時の損失関数が約 0.2 程度となっており，正しく学習されていることが確かめられた．

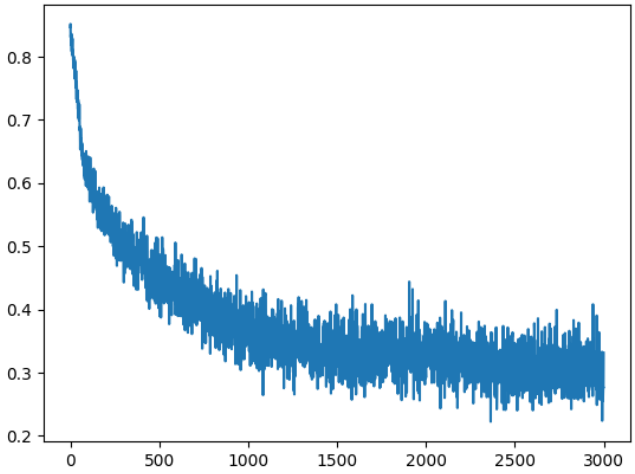

図 6: 提案手法 ($\gamma = 0$) の Stopout 検出の損失関数

次に Stopout の分類結果についてについてまとめる．Stopout の分類結果について表 5 に示す．

Stopout については，Stopout しているラベルが約 8% 程度しかなく少数ラベルについて検出が困難となる場合が多い．

分類結果について，Wheel Spinning と同様に精度，各 F 値，F 値の平均に分けて考察を行う．

まずはじめに，精度について考察を述べる．少数ラベルの損失関数に重み付けを行わない提案モデルは，従来の LSTM モデルと同じ精度であった．これは，どちらも少数ラベルの検出を行えなかったことが原因である．一方で，少数ラベルの損失関数に重み付けを行うモデルでは，少数ラベルの検出に成功しているため，精度では従来の LSTM モデルよりも低

表 5: Stopout の分類結果

評価項目 / モデル	Accuracy	0のF値	1のF値	F値の平均
提案手法(γ = 0)	0.924	0.961	NA	NA
提案手法(γ=1.0)	0.658	0.782	0.203	0.493
LSTM	0.923	0.960	NA	NA
SVM	0.917	0.957	0.002	0.479
NB	0.141	0.123	0.159	0.141
NNET	0.917	0.957	0.003	0.480
K近傍法	0.918	0.957	0.006	0.481
RF	0.917	0.957	0.007	0.482

い結果となった．提案手法の重み付けを行うモデルの方が，従来の SVM などの手法も，0 の検出が誤って検出されることが少なく，検出を行える少数ラベルの損失関数に重みを加えたモデルよりも良い精度となった．

次に各 F 値についてである．ラベル 0 の F 値について．少数ラベルの損失関数に重み付けを行わない提案モデルは，従来の LSTM モデルと同じ F 値であった．これは，0 のラベル全てが 0 と予測されており，誤りが 0 であったためである．一方で少数ラベルの損失関数に重み付けを行うモデルの場合は，Stopout の検出に重みが加わり誤りが生じるため従来の LSTM モデルよりも低い値となっている．従来の SVM などの手法と比較しても 0 の F 値にいては誤って検出される事が少ないため同程度の値となっている．

ラベル 1 の F 値について，ラベルの数が少ないためほぼ検出されない結果となった．しかしながら，少数ラベルの損失関数に重み付けを行うことで，検出が可能となり，少数ラベルの損失関数に重み付けを行う提案モデルが，Stopout を検出できなかった LSTM モデルと比べ良い結果

となった．従来のSVMなどの手法と比較しても，少数ラベルの損失関数に重みを加えたモデルがすべての中で最も高く良い結果となった．

最後にラベル0のF値とラベル1のF値を平均して求めたF値の平均についてである．少数ラベルの損失関数に重み付けをすることによりStopoutの検出が可能となったことで，従来のLSTMモデルでStopoutを検出することが出来ず平均が計算出来ないところから計算が可能となった．F値の平均では，少数ラベルの損失関数に重みを加えたモデルが従来のすべてのモデルと比べて最も高い結果となった．ラベル0のF値が低いもののラベル1のStopoutである事の検出が大事であるので，少数ラベルの損失関数に重みを加えたモデルがより優れていると考える．

以上の結果をまとめる．Stopoutの検出ではStopoutした学習者の行動数が少なく検出することが困難であったが，少数ラベルに重み付けを行う提案モデルを用いることにより，Stopoutの検出が可能となった．全体の精度は下がるものの1のF値では提案モデルが最も高く，Stopoutの検出がモデルにおいて最も大事であるので提案モデルが従来手法よりもStopoutを検出する精度おいても上回ることが明らかになった．

5　むすび

本論文では，知識状態変数を含むLSTMモデルによる新たな学習不振兆候予測の手法を提案した．具体的には以下のような手順で学習不振兆候予測を行った．

1) 学習者の問題IDとその問題の正誤からDKTを用いて隠れ層を抽出することで，知識状態変数を取り出す．

2) 従来の学習不振兆候のデータセットと抽出した隠れ層を結合することで，学習不振兆候予測の特徴量を生成する．

3) 新しく生成された特徴量を用いて学習不振兆候の予測を行う．

データセットはWheel SpinningとStopoutがタグ付けられたAssistment2016-2017を用いた評価実験により，提案手法が従来のLSTM手法やSVM，NB，NNET，K近傍法，RFよりも，Wheel Spinningにおいては精度，ラベル0のF値，F値の平均で上回り，Stopoutの検出においては，ラベル1のF値，F値の平均において上回ることができた．

モデルの少数ラベルの損失関数に重みを加えるγの値を変えることで，更に良い分類精度を目指し，さらなるモデルを発展させていきたい．

謝辞

本論文を作成するにあたり，指導教員の植野真臣教授から，丁寧かつ熱心なご指導を賜りました．ここに感謝の意を表します．また，日頃から親身になって研究を支えていただいた宇都雅輝助教に深謝いたします．そして，ゼミや日常の議論を通じて多くの示唆や知識を頂いた川野秀一准教授，西山悠准教授，研究室の木下涼先輩をはじめとする先輩方・同期・後輩に感謝いたします．

Botelho らから研究に必要な ASSistment2016-17 の提供がありました．厚く御礼を申し上げ，感謝いたします．

参考文献

[1] A. Duckworth, C. Peterson, M. Matthews, and D. Kelly, “Grit: Perseverance and passion for long-term goals,” Journal of personality and social psychology, vol.92, pp.1087–101, 07 2007.

[2] C.Peterson, A Handbook and Classification, vol.vol1, Oxford,U.K :Oxford Univ, 2004.

[3] M. Kapur, “Productive failure,” Cognition and Instruction, vol.26, no.3, pp.379–424, 2008. https://doi.org/10.1080/07370000802212669

[4] A. Botelho, A. Varatharaj, T. Patikorn, D. Doherty, S. Adjei, and J. Beck, “Developing early detectors of student attrition and wheel spinning using deep learning,” IEEE Transactions on Learning Technologies, vol.PP, 04 2019.

[5] J.E. Beck and Y. Gong, “Wheel-spinning: Students who fail to master a skill,” Artificial Intelligence in Education, eds. by H.C. Lane, K. Yacef, J. Mostow, and P. Pavlik, pp.431–440, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[6] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L.J. Guibas, and J. Sohl-Dickstein, “Deep knowledge tracing,” Advances in Neural Information Processing Systems 28, eds. by C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, and R. Garnett, pp.505–513, Curran Associates, Inc., 2015. http://papers.nips.cc/paper/5654-deep-knowledge-tracing.pdf

[7] R. Chaudhry, H. Singh, P. Dogga, and S.K. Saini, “Modeling hint-taking behavior and knowledge state of students with multi-task learning.,” International Educational Data Mining Society, 2018.

[8] N. Matsuda, S. Chandrasekaran, and J.C. Stamper, "How quickly can wheel spinning be detected?," EDM, pp.607–608, 2016.

[9] Y. Gong and J.E. Beck, "Towards detecting wheel-spinning: Future failure in mastery learning," Proceedings of the Second (2015) ACM Conference on Learning @ Scale, p.67âĂŞ74, L@S ' 15, Association for Computing Machinery, New York, NY, USA, 2015. https://doi.org/10.1145/2724660.2724673

[10] D. Chaplot, E. Rhim, and J. Kim, "Predicting student attrition in moocs using sentiment analysis and neural networks," ,vol.1432,06 2015.

[11] W. Xing, X. Chen, J. Stein, and M. Marcinkowski, "Temporal predication of dropouts in moocs: Reaching the low hanging fruit through stacking generalization," Computers in Human Behavior, vol.58, pp.119–129, 2016. http://www.sciencedirect.com/science/article/pii/S074756321530279X

[12] C. RosÃĺ, R. Carlson, D. Yang, M. Wen, L. Resnick, P. Goldman, and J. Sherer, "Social factors that contribute to attrition in moocs,",pp.197–198, 03 2014.

[13] A. Lamb, J. Smilack, A. Ho, and J. Reich, "Addressing common analytic challenges to randomized experiments in moocs: Attrition and zero-inflation,"pp.21–30, 03 2015.

[14] B.-H. Kim, E. Vizitei, and V. Ganapathi, "Gritnet: Student performance prediction with deep learning," ArXiv, vol.abs/1804.07405, 2018.

[15] A. Botelho, H. Wan, and N.T. Heffernan, "The prediction of student first response using prerequisite skills," L@S '15, pp.• •–• •, 2015.

[16] R.J. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” Neural Computation, vol.1, no.2, pp.270–280, June 1989.

[17] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol.9, no.8, pp.1735–1780, 1997.

[18] M. Khajah, R. Lindsey, and M. Mozer, “How deep is knowledge tracing?,” How Deep is Knowledge Tracing?, 03 2016.

[19] X. Xiong, S. Zhao, E.V. Inwegen, and J.E. Beck, “Going deeper with deep knowledge tracing,” EDM, 2016.

[20] B.-H. Kim, E. Vizitei, and V. Ganapathi, “Gritnet 2: Real-time student performance prediction with domain adaptation,” 09 2018.

[21] A. Botelho, R. Baker, and N. Heffernan, “Improving sensor-free affect detection using deep learning,”, 01 2018.

[22] A. Sales, A. Botelho, T. Patikorn, and N.T. Heffernan, “Using big data to sharpen design-based inference in a/b tests,” EDM, 2018.

[23] C.K. Yeung, Z. Lin, K. Yang, and D.-y. Yeung, “Incorporating features learned by an enhanced deep knowledge tracing model for stem/non-stem job prediction,” 06 2018.

[24] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” IEEE Transactions on Neural Networks, vol.5, no.2, pp.157–166, March 1994.

[25] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” International Conference on Learning Representations, Dec. 2014.

[26] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Z. Zhang, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3982–3991, June 2015.

[27] C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol.20, no.3, pp.273–297, 1995.

[28] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," Machine learning, vol.29, no.2-3, pp.131–163, 1997.

[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.580–587, 2014.

[30] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al., "Constrained k-means clustering with background knowledge," Icml, vol.1, pp.577–584, 2001.

[31] L. Breiman, "Random forests," Mach. Learn., vol.45, no.1, p.5âĂŞ32, Oct. 2001. https://doi.org/10.1023/A:1010933404324