Jh210019-NAH

機械学習ソフトウェアへのソフトウェア自動チューニング技術の適用

田中 輝雄（工学院大学）

概要
ソフトウェア自動チューニング(AT)におけるユーザプログラムの性能を決定する性能パラメタの最適組合せ探索に関する研究を進めている．ここではロボット制御における人移動の予測モデルを，機械学習を用いて，より精度の高い予測モデルに順次修正する問題を扱う．このときの機械学習のハイパーパラメタ最適化を AT で行う．機械学習では膨大な学習時間を要し，さらに，大量のデータ利用にランダム性があり，結果が一意に定まらない．以上のような課題に対し，(1)名古屋大学 不老 TypeII の持つ多数の GPU を有効に用いた多重実行手段を構築した．並列化により，AT のハイパーパラメタ探索時間を逐次実行に比べて 13-23 倍高速化した．(2)機械学習で生じる揺らぎに対応するため，得られた上位複数の結果を再度実行し，学習結果の安定性を確保した．その結果，評価指標である人移動の最終位置の誤差に対し，従来の値から 29-38%改善した学習モデルを構築できた．

1. 共同研究に関する情報

(1) 共同研究を実施した拠点名

名古屋大学

(2) 共同研究分野

超大規模数値計算系応用分野

(3) 参加研究者の役割分担

田中輝雄（工学院大学・情報学部）
・取り纏め，自動チューニング手法
大島聡史（名古屋大学・情報基盤センタ）
・GPU，メニコア向け，アルゴリズム
藤井昭弘（工学院大学・情報学部）
・並列処理アルゴリズム
加藤由花（東京女子大学・現代教養学部）
・機械学習プログラム・データ提供
藤家空太郎(工学院大学・大学院情報学専攻)
・自動チューニングソフトウェア開発，評価
嵯峨山凌(工学院大学・大学院情報学専攻)
・自動チューニングソフトウェア開発，評価
ヨウケン(工学院大学・大学院情報学専攻)
・自動チューニングソフトウェア開発，評価

2. 研究の目的と意義

ソフトウェア自動チューニング(AT)では，ユーザプログラムの性能を決定する複数性能パラメタを操作することで，ユーザプログラムのチューニングを行う．我々はこの複数の性能パラメタの最適組合せ探索の研究を進めている[a]．本研究では「人を回避しながら動くロボット制御」に対する歩行者移動予測 AI[b]の機械学習プログラムのハイパーパラメタの組合せの選択に，我々の AT 技術を適用する．

対象とするシステムと我々の研究の位置づけに関して図 1 に示す．対象システムは Physical world と Cyber World(クラウド)から構成される．

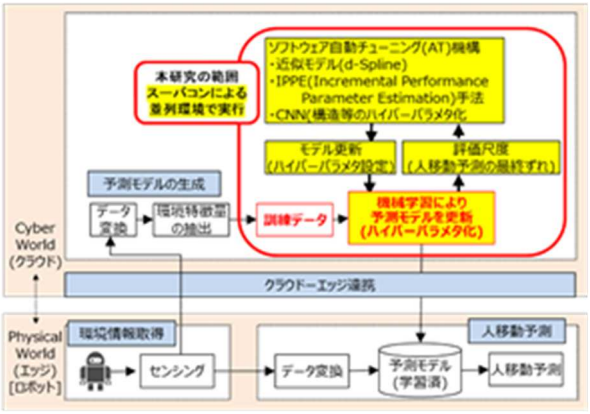


図 1 対象システムと本研究の位置付け

Physical world では，ロボットが予測モデルを用いて周囲の人移動を予測し，自立動作を行う．ロボット動作時のセンシング情報を随時 Cyber World(クラウド) にフィードバックし，バックグラウンドで予測モデルを機械学習により更新する．

この Cyber World(クラウド)上での機械学習において，その機械学習のハイパーパラメタの AT を行う．機械学習では大量のデータを扱い，一般に GPU を用いた膨大な学習時間を要する．さらに，学習用の対象データを選択することにより，学習内容が異なるので，学習結果が一意に定まらない．

これらの状況から，本研究では，以下に示す２項目を目的とする．

【目的１】スーパーコンピュータの持つ多数のプロセサから成る高性能環境を有効に利用し，並列化によりハイパーパラメタの探索時間を大幅に削減すること．

【目的２】機械学習において生じる学習結果の揺らぎを評価し，推定したパラメタの安定性を確保することにより,AT の実行結果の信頼性を高めること．

なお，本研究は，ロボット工学の専門家と高性能計算学の専門家との学際研究として推進している．

[a]T.Tanaka, R.Otsuka, A.Fujii, T.Katagiri, T.Imamura, “Implementation of d-Spline-based Incremental Performance Parameter Estimation Method with ppOpen-AT,” Scientific Programming, Vol.22, pp.299-307, 2014.

[b]R.Akabane and Y.Kato, “Pedestrian Trajectory Prediction Using Pre-trained Machine Learning Model for Human-Following Mobile Robot,” IEEE International Conf. on Big Data Workshop (IoTDA 2020), pp.3453-3458, 2020.12.

3. 当拠点公募型研究として実施した意義

本研究課題で対象とする機械学習は膨大な学習時間を要する.AT ではこの膨大な計算に対してハイパーパラメタを変更しながら，ユーザプログラムを何度も実行する必要がある．また，機械学習においては計算の特性から GPU が用いられる．そのために，多数の GPU を有するスーパーコンピュータである　名古屋大学情報基盤センターの不老 Type II （以下，不老 Type II)を用いる．本研究では，この不老 Type II の持つ，大規模 GPU 並列環境を有効に利用する手段を提案し，実証することにより，この分野に対する貢献を行う．

4. 前年度までに得られた研究成果の概要

前年度はありません

5. 今年度の研究成果の詳細

(1)簡易モデルでの試行

GPU を用いた１回あたりの予測モデルの機械学習時間が 30 分程度かかる簡易モデルを用いて，我々の AT 技術を試行した．なお，ここで今回，機械学習で更新する予測モデルの改善の指標は，人移動予測による最終位置のずれ幅(FDE: Final Destination Error)としており，このずれ幅をできるかぎり小さくすることを目指す．

この機械学習プログラムを,AT で自動選択したハイパーパラメタの組合せ（厳密には，ハイパーパラメタの取りうる値の組合せ，以下同様）を変更し繰り返し実行する．簡易モデルでは，対象を主要なハイパーパラメタ 6 種とし，それぞれの取りうる値を５通りとした.したがって，ハイパーパラメタの組合せは 5^6=15k パタンとなる．

不老 Type II を用いて，簡易モデルに対してハイパーパラメタの値の組合せの推定を行った．結果を表１に示す．

	実行時間	探索回数	FDE(m)
逐次実行	279時間（11.6日）	553回	0.89
並列実行	12.3時間	761回	0.81
初期パラメータ			1.14

1/23

表１ 並列化による効果と解精度の向上

対象の機械学習プログラムを 553 回実測し探索を行うことにより 11.6 日を要した.そこで，不老 Type II の複数 GPU ノードを有効に用いた機械学習プログラムの多重実行により，それぞれ異なるハイパーパラメタの組合せによる探索を並列に実行する（いわゆるパラメトリックスタディを行う）.このことにより，大幅に時間を削減することを可能とし，全体で 12.3 時間と約 1/23 に短縮することができた．多重実行では，後述するように逐次実行に比べて冗長な実測を行うため，対象の機械学習プログラムの実行回数は 761 回と増加している.

この推定したハイパーパラメタの値の組合せを用いた予測モデルにより，現状用いられているハイパーパラメタの組合せで学習した予測モデルに比べて 29%(=0.81/1.14)ほど学習結果の良いハイパーパラメタの値の組合せを得た（以上【目的１】に対する試行）.

また，機械学習は確率的要素があるので，動作するたびに解に変動する可能性がある.

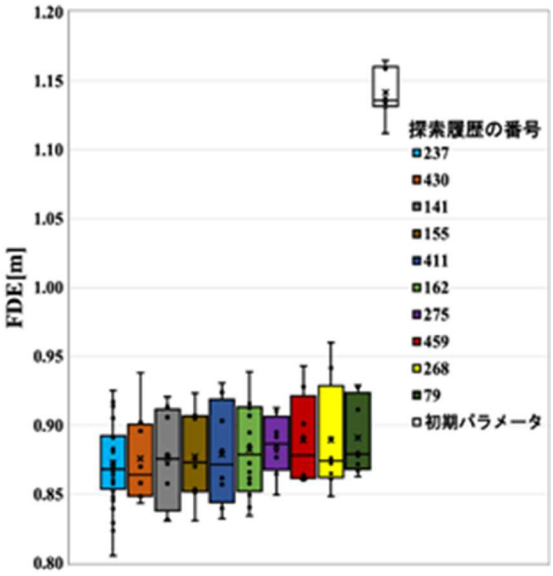


図２ 学習による評価値 FDE の揺らぎ

今回の AT にて得られたハイパーパラメタの組合せからなる学習した予測モデルにより，評価指標 FDE がどのくらい揺らぐかを調べ，選択したパイパーパラメタが安定した結果を出すかを調査した．ここでは，求められた FDE の中から上位 10 番目までの値が良い(つまり FDE 値が小さい)ハイパーパラメタの組合せについて，再度それぞれ 10 回ずつ機械学習を実行した.

結果を図２に示す．各ハイパーパラメタの組合せによる FDE のばらつきは,初期値(つまり，従来用いられているハイパーパラメタの組み合せ：図中の右端の白枠）に比べて，明らかに有意な範囲でのばらつきで抑えられていることが確認できた．これにより，今回得られたハイパーパラメタの組合せは有効と判断することができる（以上【目的２】に対する試行）.

(2) 適用先環境データに対する機械学習

ロボットは様々な環境で自走することが求められる．そのような様々な環境では，歩行者の混雑具合や行動パタンなどに違いがあり，それぞれの最適な予測モデルは異なる可能性が考えられる．ここでは，“hotel”および“bookstore”の２つのデータを用いて実験を行った.

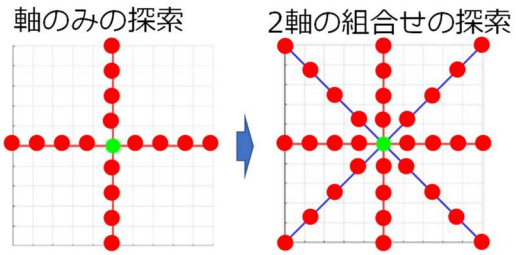
(2-1) データ“hotel”

実験結果を表２に示す.

ハイパーパラメタの組合せを変更しながら，205 回の学習を行い，１回あたりの平均学習時間は 20 分 40 秒であった．これを不老 Type II の並列環境で実行することで全体の実行時間は約 4 時間となった.なお,不老 Type II では，運用形態から 50 ジョブまで同時に実行できる.つまり，今回の多重実行度は平均約 17.1(=1,240×205/14,429)となっており，まだ多重度には余裕がある.

表２ データ“hotel”に対する実験結果

実験データ(hotel)		並列実行	
		最適値	初期値
ハイパーパラメタ	run_size	512	128
	grad_chip	10	10
	Learning_rate	0.003	0.003
	dropout	0.3	0.5
	Lambda_param	0.002	0.0005
FDE		1.14	1.85
実行時間		14,429秒(4時間29秒)	ー
学習回数		205	
平均時間／学習		1240秒 (20分40秒)	



補助図 並列探索対象範囲（２次元の場合）

ここで，AT の初期値として用いた値は，これまで最適と考えて使われていたハイパーパラメタの組合せである．この初期値と AT で推定されたハイパーパラメタの組合せには差異が生じた．これにより，評価に用いた FDE は，1.85[m]から 1.14[m]となり，今回の AT により 39%の改善が見られた．

この AT の動作を分析する前に，補助図を用いて我々の AT 方式のポイントを概説する．

我々のハイパーパラメタ推定は，複数のハイパーパラメタの取りうる値からなる多次元の(離散)空間を考え，その空間内で探索を行う．指定した初期点からはじめ，その初期点からまず各軸方向のみの各点を評価し，この範囲でコスト(今回は FDE)が最小となる点を見つけ，その点を移動させて，同じ処理を実施する．点の移動がなくなると，軸のみの探索から探索範囲を拡大し，２軸の組合せとなる斜め方向のライン上の点を追加して評価する．この組み合わせる軸の数を順次増やして斜めのラインを作り上げることで，探索領域を段階的に広げていく．なお，一度実測済みの点の再測定は行わない．

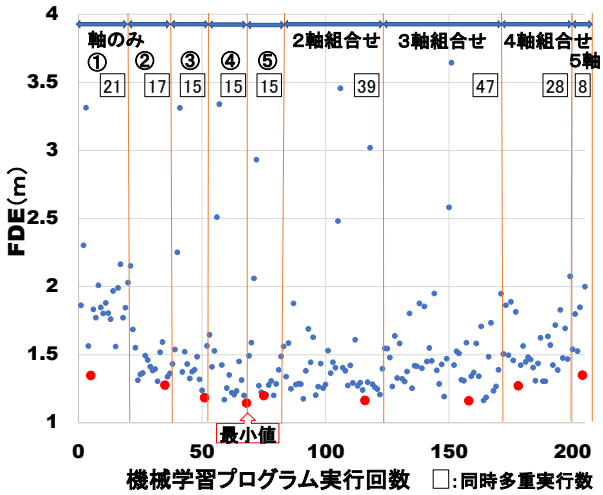


図３ AT 実行状況（データ：hotel）

今回の多重ジョブの並列実行は，この２軸方向の各点，あるいは，斜め方向も含めた各点を同時に行っている．今回の機械学習プログラムのハイパーパラメタは５種であり，対応した探索するハイパーパラメタ空間は５次元となる((1)簡易モデルから１種減)．なお，各々のハイパーパラメタの取りうる値は５通りである．

AT 実行状況を図３に示す．図中，青点のひとつひとつが機械学習プログラムの実行結果として得られた FDE 値である．橙の縦線の間が同時多重実行の範囲であり，□内の数字は同時多重実行数である．最初は５本の軸上の点をすべて実行し，その中で最小の FDE の点を見つける．その点を次の基準点とする．各多重実行での最小 FDE の点を図では赤点で示している．②～⑤はこの赤点を基準に各軸を探索する（②～⑤の段階はまだ軸方向のみとなる）．なお，すでに実行している点については実行しない．５回目では４回目で見つけた最小値の方が，５回目で新たに実測したすべての値より小さいので基準点は変わらない．このタイミングで斜め方向の探索ラインを増加している．

今回の場合は，探索ラインをさらに増やしていっても FDE が最小となる点は更新されていない．軸間を組み合わせたラインを増やしても新たな最小値が見つからなかったことはハイパーパラメタ間の相関が弱いとも考えられる．

表３ 特異値分解（データ：hotel）

特異ベクトル						特異値
rnn_size	grad_clip	learning_rate	dropout	lambda_param	FDE	
0.581	0.000	0.045	-0.394	0.202	-0.681	19.6
-0.438	-0.135	0.303	-0.476	0.675	0.122	15.3
0.119	-0.779	0.556	0.172	-0.202	-0.022	14.4
0.050	0.613	0.773	0.110	-0.113	-0.002	14.1
-0.241	-0.005	0.022	-0.706	-0.666	0.006	12.7
-0.629	-0.002	-0.010	0.279	-0.076	-0.721	6.8

この相関についてはさらに分析を必要と考えている．

次にハイパーパラメタの特性を分析するために，特異値分解を行った．結果を表３に示す．この結果から，特異ベクトルの rnn_size，dropout，lambda_param の項が大きいことがわかる．実際，今回は最初に，rnn_size の値が，さらに lambda_param，dropout の順に基準点が移動した．この特異ベクトルについては，今後，分析を行い，さらに効率の良い AT の実現につなげていく．

(2-2) データ“bookstore”

データを“bookstore”としたときの AT 実験を行う．実験結果を表４に示す．

ここで，ハイパーパラメタは“hotel”と同じように５種類であり，それぞれが５つの値を取る．“bookstore”では１回あたりの学習時間が 5.57 時間と大幅に長くなっている．今回の AT のハイパーパラメタを変更したときのユーザプログラムの実行回数は 170 回となった．

つまり，仮にこれを逐次実行した場合は，推定で 39 日程度(= 5.57×170/24 = 39.4)の時間を要すると考えられる．

なお，評価指標である PDE も初期値 0.71[m]に比べて 0.49[m]と 31%(=0.49/0.71)の改善が得られた．

今回の AT の詳細な動作を図４の AT 実行状況に示す．表示の形式は(2-1)“hotel”（図３）と同じである．今回も“hotel”と同様に，１軸のみの時点で見つかった値が最適値と判断されている．

表４ データ“bookstore”に対する実験結果

実験データ(bookstore)		提案AT並列実行		参考(乱数)
		最適値	初期値	最適値
ハイパーパラメタ	run_size	512	128	512
	grad_chip	7	10	12
	Learning_rate	0.003	0.003	0.001
	dropout	0.3	0.5	0.3
	Lambda_param	0.0001	0.0005	0.002
FDE		0.4872	0.7081	0.5085
実行時間		2.4[日]	―	21.6[時間]
学習回数		170		170
平均時間／学習		20051秒 (5.57時間)		―

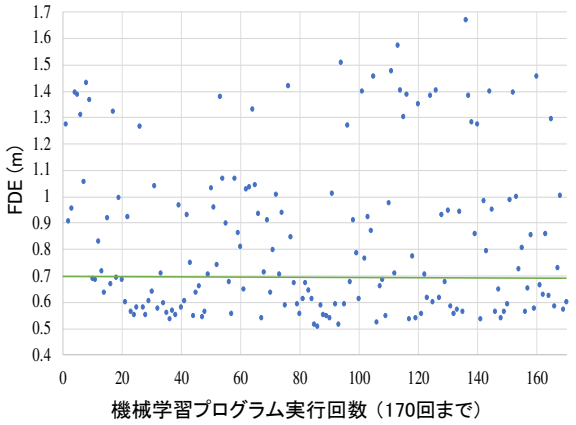


図４ AT 実行状況（データ：bookstore）

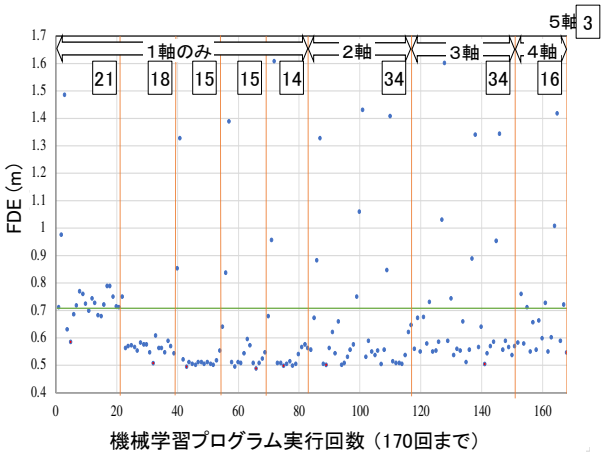


図５ AT 実行状況　乱数を用いた探索
（データ：bookstore）

ここで，我々の提案した AT(以下，提案 AT)の効果を示すために，ハイパーパラメタの組合せの設定に乱数を用いた場合について実験を行った．乱数での実行回数は提案 AT で実行した 170 回に合わせた．このときの　結果を表４の最右列に，また，挙動について図５に示す．

まず，表４から推定した FDE の最適値は，0.51[m]となった．一方，提案 AT は 0.49[m]．また，実行時間は提案 AT が 2.4 日かかっているのに対して 21.6 時間で終わった．これは，提案 AT が探索を進めているのに対して，乱数の方は，各ハイパーパラメタの実行において，互いに依存関係がないために，不老 TypeII の並列実行の上限(=50)を有効に利用できるためである．提案 AT においては，更なる並列化の効率化が必要となる．

図４と図５を比べてみると，相対的に図４の提案 AT の方が乱数での実行より良い値を取る点を探索しているのがわかる．実際，初期値の FDE 値(0.71[m])より小さなハイパーパラメタの組合せを調べたのは，提案 AT で 75%，一方，乱数では 48%となっている．ただ，これは見方を変えると探索の局所性が強いとも言えるため，さらに大域的な探索機能も考慮する必要もあると考える．

6. 今年度の進捗状況と今後の展望

本報告では，ロボット制御における人移動の予測モデルを機械学習により，より精度の高い予測モデルに順次修正する．この機械学習のハイパーパラメタの最適化をソフトウェア自動チューニング（AT）で実現する．

これまでに，機械学習における膨大なデータ処理を AT として繰り返し実行するために，不老 Type II の持つ多数 GPU を有効に用いることを可能とした．また，機械学習における大量データの入れ替えによる結果の揺らぎに対し，得られた上位の複数の結果を再度実行し，評価することで，安定した結果を評価できることを確認することができた．

これらの技術を複数の異なるロボットの動作環境のデータに対し適用し，それぞれの最適なハイパーパラメタの組合せを AT で探索した．その結果，現在の予測モデルより，人移動予測による最終位置のずれ幅(FDE)を評価指標として，さらに高い精度を持つ予測モデルを構築することができた．

今後は，以下の項目について，検討を進めていく．

(a)異なる環境での予測モデルの利用：

ここでは，異なる環境での予測モデルを　初期値として，新しい環境の予測モデルを構築する効果について検討する．

(b)適用先アプリケーションの拡大

超解像プログラムの適用の検討を開始している．対象の超解像プログラムは，１回の学習において９時間ほど有する．このままでは AT は実現できないので，学習時間の削減，および，今回実現した多重実行環境の適用を行うことにより，AT の実現を図る．

(c)更なる並列利用効率の向上：

計算機環境の運用並列度を意識した多重実行管理機能について検討する．

(d)不老 TypeII 以外の HPC 環境への適用：

今回開発した多重実行環境の他システムへの適用を行う．

(e)安定性の確保から探索の精度の向上：

最終段階でのチェックだけでなく，探索途中の精度向上を行うことで，さらに良い解の抽出を目指す．

7. 研究業績一覧

(1) 学術論文 （査読あり）

なし

(2) 国際会議プロシーディングス（査読あり）

なし

(3) 国際会議発表 （査読なし）

[1] Sorataro Fujika, Akihiro Fujii, Teruo Tanaka, Yuka Kato, Satoshi Ohshima, Takahiro Katagiri, Parallelization of Automatic Tuning using Multiple Jobs for Machine Learning Programs, the 20th International Symposium on Advanced Technology (ISAT-20), 2021.11.

[2] Sorataro Fujika, Teruo Tanaka, Akihiro Fujii, Yuka Kato, Satoshi Ohshima, Takahiro Katagiri, Parallelization of Automatic Tuning using Multiple Jobs for Machine Learning Programs, Parallelization of Automatic Tuning using Multiple Jobs for Machine Learning Programs, 2021.10.

(4) 国内会議発表 （査読なし）

[3] 矢島雄河，森戸建太郎，藤家空太郎，楊 暄，藤井昭宏，田中輝雄，ソフトウェア自動チューニングにおける反復 1 次元 d-spline 探索法の複数か所同時適用，情報処理学会第８４回全国大会，2022.3.

[4] 森戸建太郎，矢島雄河，藤家空太郎，楊 暄，藤井昭宏，田中輝雄，ソフトウェア自動チューニングツール DSICE の構築，情報処理学会第８４回全国大会，2022.3.

[5] 藤家空太郎，藤井昭宏，田中輝雄，加藤由花，大島聡，不老上での自動チューニング機構の並列化と機械学習プログラムに適用時の安定性の検証，第２回スーパーコンピュータ「不老」ユーザ会，2021.10.

[6] 田中輝雄，大島聡，藤井昭宏，滝沢寛之，多部田敏樹，藤家空太郎，機械学習ソフトウェアへのソフトウェア自動チューニング技術の適用，2020 年度(令和 2 年度)名古屋大学 HPC 計算科学連携研究プロジェクト成果報告会，2021.7.

(5) 公開したライブラリなど

なし

(6) その他(特許，プレスリリース，著書等)

なし