小口研究室　研究紹介　(2018年度)
(お茶の水女子大学理学部情報科学科)

動画像データを用いた機械学習による動作識別手法の比較
（研究担当:高崎 智香子）

研究背景

防犯カメラなどの動画像データの活用
ディープラーニング技術の応用
- 正確な識別処理には大量のデータ収集・処理が必要
- リアルタイムに識別処理を行うのは非常に困難

➡ センサとクラウドで識別処理を分散

大量の動画像データを使用して機械学習処理を行うことによって計算量が膨大になる

動画のままセンサからクラウドに送信すると通信量やプライバシーの問題が生じる

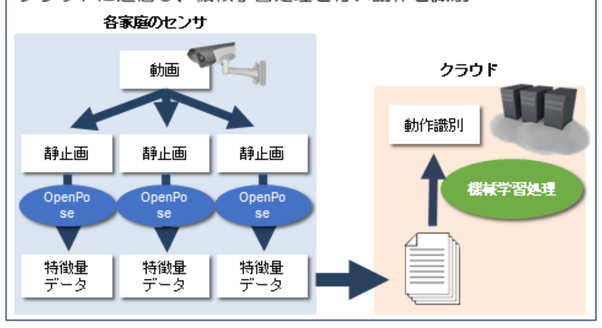
センサ側における前処理、
クラウド側における識別処理の並列化
によって負荷分散

OpenPose
- 姿勢推定ライブラリ
- 135の特徴点を認識可能
- 特殊センサを使わずに解析

Keras
- NN実装ライブラリ
- GPUで高速に動作
- モデルの記述が容易

研究概要

各家庭のセンサで収集した動画像データを静止画に変換後、OpenPoseを用いて特徴量データに変換。特徴量データのみをクラウドに送信し、機械学習処理を行い動作を識別



実験概要

動画像を画像に変換
➡ 画像から特徴量を取得
➡ 機械学習処理

STAIR Actions※の動画を画像に変換
OpenPoseでキーポイントを解析
複数の手法で動作の識別精度を比較

実験結果

2種類のデータを使用して以下について調査
a. 画像 1 枚のキーポイントの座標データ
b. 画像 10 枚の時系列を考慮したキーポイントの座標データ

1. 各機械学習手法による識別精度の比較
   - ロジスティック回帰
   - ランダムフォレスト
   - SVM
   - NN
2. NNにおけるパラメータ調節
   - 中間層の層数とノード数
   - DropoutとBatchNormalization

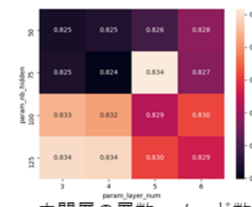
1. 各機械学習手法による識別精度の比較

	a. Training	a. Test	b. Training	b. Test
ロジスティック回帰	0.688	0.640	0.869	0.580
ランダムフォレスト	1.000	0.786	1.000	0.828
SVM	1.000	0.454	1.000	0.440
NN	1.000	0.828	0.976	0.748
NN w/Dropout	0.987	0.820	0.999	0.800
NN w/BN	1.000	0.842	0.999	0.813
NN w/Dropout, BN	0.970	0.813	0.987	0.765

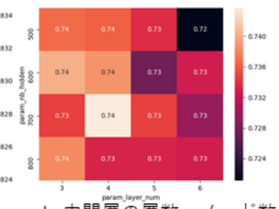
a. NN
b. ランダムフォレストで最も精度良い

- NNでは過学習の傾向
- a→b精度低下
- Dropout/BNで精度改善

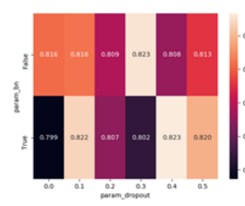
2. NNにおけるパラメータ調節



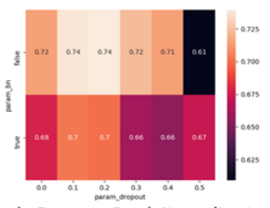
a. 中間層の層数・ノード数



b. 中間層の層数・ノード数



a. Dropout, Batch Normalization



b. Dropout, Batch Normalization

考察
- NNの学習の様子から、損失が下がっていないため、標準化・正規化手法を見直す必要がある
- aと比較してbでは動作の特徴を捉えやすいが、データ数が10分の1になっているため精度が下がっている可能性

まとめと今後の課題

まとめ
- 動画をOpenPoseで前処理した後、機械学習で動作識別
- ２種類のデータで複数手法による識別制度を比較

今後の課題
- LSTMによる動作識別
- センサ、クラウドによる分散環境への実装

機械学習向けのコンピュータシステムの構築に向けたAIワークロードの特徴分析
（研究担当:高山　沙也加）

研究背景

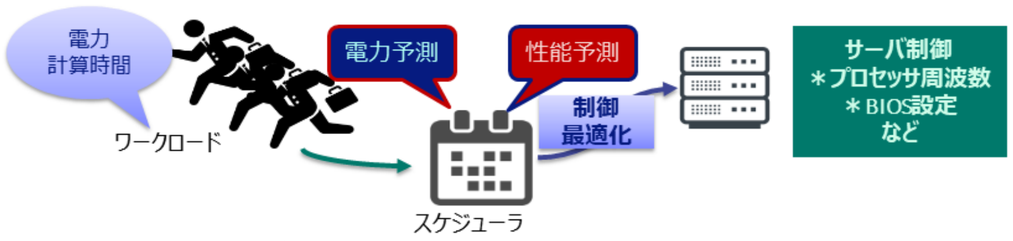
◆ クラウドコンピューティングによる解析やAIを用いたアプリケーションの利用の増加
  ◆ GPU利用の増加により
  ICTシステムの全体電力は増加傾向
◆ AIワークロードを走行させる
ハードウェアリソースの有効活用・運用の手法は未だに確立されていない

ワークロード：
計算機資源の利用状況を示す指標
  ◆ 分析を元にシステムを効率よく稼働させることで，
  コストパフォーマンスの向上を図る

研究目的

◆ ハードウェアを有効活用することで
AIワークロードを高速化, 効率化する運用技術の確立

具体的には...
ワークロード毎にサーバ性能の自動チューニングを行うコンピュータシステム

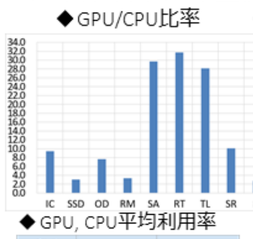


◆ AI処理時のサーバやプロセッサの情報をアプリケーション毎に特徴分析
◆ ハードウェア構成が異なる条件でのワークロードを解析

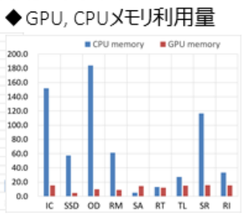
実験・分析結果

◆ MLPerf
大学・研究機関が連携して開発している機械学習分野のベンチマーク
一通りの分野が網羅されているためベンチマークとして利用
◆ Zabbix
サーバやネットワーク監視用のソフトウェア

◆GPU/CPU比率



◆GPU, CPUメモリ利用量

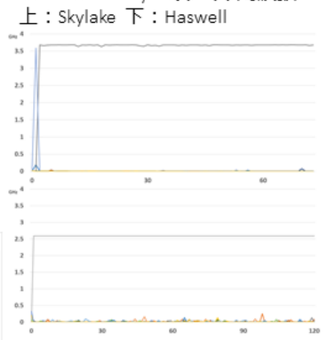


◆GPU, CPU平均利用率

ベンチマーク	CPU利用率(%)	GPU利用率(%)
IC	5.1	95.4
SSD	9.9	59.8
OD	4.9	74.5
RM	6.7	44.3
SA	1.4	82.0
RT	1.5	95.5
TL	1.5	83.9
SR	3.2	65.1
RF	11.4	62.7

全体的にGPUネックだが分野毎にある程度傾向が見られた
GPUの容量不足を解決できればより効率の良い運用が可能に

◆ Sentiment analysisのクロック周波数
上：Skylake 下：Haswell



CPU変更によるクロック周波数がアプリケーション性能に影響を及ぼすと推測
GPU平均利用率が高いベンチマークほどGPUを変更した際のジョブ実行時間の変化が大きく、GPU利用率が高いジョブはGPU性能によるジョブ性能向上効果が高いと考えられる

◆ ジョブ時間 – サーバ比較

ベンチマーク	Haswell (h:min:s)	Skylake (h:min:s)	Skylake / Haswell
IC	4:36:10	4:33:04	0.99
SSD	3:56:42	3:12:18	0.81
OD	2:59:34	2:57:51	0.99
RM	1:12:00	1:09:07	0.96
SA	2:00:12	1:48:14	0.90
RT	4:06:41	4:05:28	1.00
TL	4:37:47	4:29:21	0.97
SR	-	15:07:34	-
RF	6:28:00	6:08:37	0.95

◆ ジョブ時間 – GPU比較

ベンチマーク	P100 (h:min:s)	V100 (h:min:s)	P100 / V100
IC	4:33:04	3:01:13	1.51
SSD	3:12:18	3:09:12	1.02
OD	2:57:51	2:23:26	1.24
RM	1:09:07	1:09:34	0.99
SA	1:48:14	1:22:50	1.31
RT	4:05:28	2:48:18	1.46
TL	4:29:21	2:59:55	1.50
SR	15:07:34	10:53:32	1.39
RF	6:08:37	5:46:00	1.07

今後の課題

◆ ジョブとリソースの関係のより詳細な調査
◆ 実環境アプリケーションや複合ベンチマークを単純ベンチマークの複合情報で再現可能であるか検証
◆ 正解データやデータ量からGPU，メモリ量を自動で選択するコンピュータシステムの構築

完全準同型暗号を用いたFP-growthによる頻出パターンマイニングの分散処理への実装
（研究担当:種村　真由子）

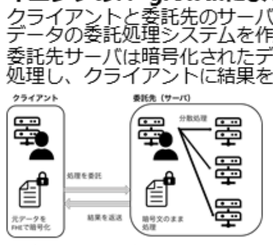
研究背景

- **ビッグデータなどの利活用**
  IoT・医療等、様々な技術が大量のデータの収集・活用により
  発展するという期待がある
- **大規模データ処理の外部委託**
  大規模なデータ処理には高性能な計算機が必要となるため、外部に計算を委託するのが現実的
- **機密情報のセキュリティ管理の必要性**
  信用できない外部の業者には、機密データの処理を依頼できない

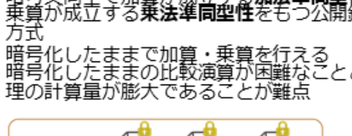
→外部に元データを公開せずに委託処理を行うことができれば理想的

提案手法

完全準同型暗号を用いた頻出パターンマイニングのFP-growthによる実装
クライアントと委託先のサーバを想定した、データの委託処理システムを作成する
委託先サーバは暗号化されたデータを復号せず処理し、クライアントに結果を返送する



●完全準同型暗号
暗号文同士で加算が成立する**加法準同型性**と、乗算が成立する**乗法準同型性**をもつ公開鍵暗号方式
暗号化したままで加算・乗算を行える
暗号化したままの比較演算が困難なことと、処理の計算量が膨大であることが難点



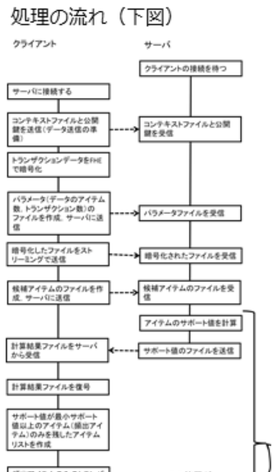
●頻出パターンマイニング
トランザクションの集合から、一定以上の頻度で出現するパターンを抽出する手法

●FP-growth
頻出パターンマイニングのアルゴリズムの1つ
FP-treeというprefix treeのデータ構造を用いた、深さ優先探索型のアルゴリズム
<長所>
・頻出パターンを列挙しない
・データベースのスキャンが2回で終了
・データが更新された時に、処理をはじめからやり直す必要がない

<短所>
・FP-treeをメモリ上に保存するため、メモリの使用量が多くなる傾向がある

先行研究において、完全準同型暗号を用いた、Aprioriという別手法による頻出パターンマイニングとその高速化が実現しているが、本研究では、頻出パターンマイニングの**アルゴリズムをApriori→FP-growthに変更**したプログラムを実装する

処理の流れ（下図）

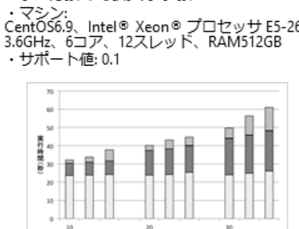


実装・実験

実装
- 主な使用言語：C++
- 使用ライブラリ
  - Boost
  - Helib（FHEライブラリ）
  - OpenMPI（MPIライブラリ）

実験1
client、masterプログラムの実行時間を計測

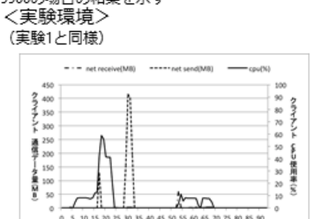
<実験データ>
IBM Quest Synthetic Data Generatorにより作成した人工データ
・アイテム数: 10, 20, 30
・トランザクション数: 3300, 6600, 9900
<実験環境>
・ワーカ数1、使用マシン数2
・マシン:
CentOS6.9、Intel® Xeon® プロセッサ E5-2643 v3、3.6GHz、6コア、12スレッド、RAM512GB
・サポート値: 0.1



実験2
client、masterプログラムを動かした際の使用リソースを測定

<実測定項目（測定にはdstatを使用）>CPU使用率（%）、メモリ使用量（MB）、ネットワークの送受信データ容量（MB）

<実験データ>
（実験1と同様に作成）
結果にはアイテム数: 30、トランザクション数: 9900の場合の結果を示す
<実験環境>
（実験1と同様）



実験結果より
実験1より、トランザクション数の変化と比較して、アイテム数の変化による実行時間の影響が大きい
実験2より、クライアント、サーバどちらにおいても、通信時のリソース使用量が大きい（暗号の処理が重いためであると考えられる）
データの送受信回数やデータの容量はマシンの負荷に大きく影響することがわかったため、今後の追加実装でも考慮が必要

今後の課題

- FP-growthのTree構築以降の処理におけるサーバ委託部分を増加させる方法の検討
- クライアント、サーバ間の処理の割り振りのバランスを考える
- FP-growthの走査部分の改善を行い、より大きなアイテムセットの処理にも耐えうるようにする

http://ogl.is.ocha.ac.jp/
oguchi@is.ocha.ac.jp