7T-01

Android端末における時系列データ予測モデルの性能評価

佐藤　里香†　　山口　実靖††　　神山　剛†††　　小口　正人†
†お茶の水女子大学　　††工学院大学　　†††長崎大学

1　はじめに

近年, TensorFlow Liteなどのように機械学習のモデルをスマートフォン等の端末に組み込み, 推定処理を端末内で完結させる環境が整備されつつあり, 推定処理にリアルタイム性が求められるアプリへの活用が期待されている.

本稿では, Android端末上でトラフィックの輻輳を予測して輻輳制御を行うシステムを想定し, そのようなアプリ実装形態の実現可能性を検証すべく, 予め高性能なサーバで端末におけるトラフィックデータを学習した深層学習モデルを, TensorFlow Liteにより検証用のAndroidアプリに組み込み, その性能を検証し実現可能性を評価する.

2　関連研究

2.1　TensorFlow Lite

TensorFlow Lite[1]はGoogleの機械学習向けソフトウェアライブラリであるTensorFlowのモバイル環境向けのライブラリである. TensorFlow Liteでは, TensorFlowにより学習されたモデルをTFLite Converterを使ってTensorFlow Lite形式に変換し, デバイスに組み込むことによりデバイス上で推論を行うことができる.

2.1.1　TFLite Converter

TFLite Converterは機械学習モデル形式の変換を行う変換器である. サーバ側でTensorFlowにより機械学習を行ったのち, その学習モデルをTFLite Converterにより TFLite FlatBuffer Fileに変換する.

2.1.2　TFLite Interpreter

サーバで生成されたTFLite FlatBuffer Fileをクライアント側でTFLite Interpreterを用いることによりデバイス上で利用することができる. 本研究ではこのTensorFlow Liteを用いて時系列データ学習モデルをAndroid端末に組み込み利用する.

2.2　輻輳制御ミドルウェア

本研究で取り扱うアプリ実行形態が対象とする事例の一つとして輻輳制御ミドルウェア[2]と呼ばれるツールが挙げられる. このツールは複数台のAndroid端末が同一の無線LANアクセスポイントに接続する際に, 端末間で輻輳の状態を把握し, 協調して輻輳を制御するということを目的としている.

このミドルウェアを用いて輻輳制御するには, 輻輳の予兆を端末側で検知することが必要となり, 過去検討では深層学習を用いた手法が提案されている[3]. そこで本稿では, その具体的な実現手段として, TensorFlow Liteを適用した学習モデルを端末に組み込み, 端末上で予測処理を実行させる.

3　学習モデルの作成

本章において, 検証用アプリに組み込む深層学習モデルを作成する. まず3.1においてサーバ上でトラフィックデータを重回帰分析により学習させる. 続いて3.2でその学習モデルをTensorFlow Liteにより端末に組み込める形式に変換する.

3.1　TensorFlowモデルの作成

2.2で述べた輻輳制御システムでは, 同一アクセスポイントに接続された端末数とRTTの増減比率から適切な輻輳ウィンドウの補正値を設定している. 本研究が想定するシステムでは, 輻輳を検知して制御するのではなく輻輳発生前にその予兆を捉える必要があるため, 輻輳を示す指標としてスループット値を採用しその変化を予測する.

本実験では, Android端末5台のパケット通信時のスループットデータを, サーバ上において重回帰分析を用いて学習させた. 入力データをt - 10秒からt - 1秒まで10秒間のスループットの値とし, この入力データからt秒におけるスループットの値を予測した. また, 計181秒間の通信のうち, 7割を学習データ, 3割をバリデーションデータとして学習を行った. この学

Performance Evaluation of Time Series Data Prediction Model on Android Devices
†Rika Sato　††Saneyasu Yamaguchi　†††Takeshi Kamiyama　†Masato Oguchi
†Ochanomizu University
††Kogakuin University
†††Nagasaki University

習モデル (TF モデル) を 3.2 で端末に組み込める形式に変換する.

3.2 TensorFlow Lite モデルの作成

3.1 にて作成した TF モデルを 2.1 の流れに沿って TFLite Converter を用いて TensorFlow Lite 対応形式に変換を行った. この変換されたモデル (TF Lite モデル) をアプリとして端末に組み込む.

4 TF Lite モデルの性能評価

4.1 深層学習モデルを組み込んだアプリケーション

3.2 にて作成した TF Lite モデルを組み込んだ図 1 のような Android アプリ (検証用アプリ) を実装した. この検証用アプリを用いて, TF Lite モデルによる端末上での予測性能を, 予測精度, 予測時間, CPU 使用率について評価する.

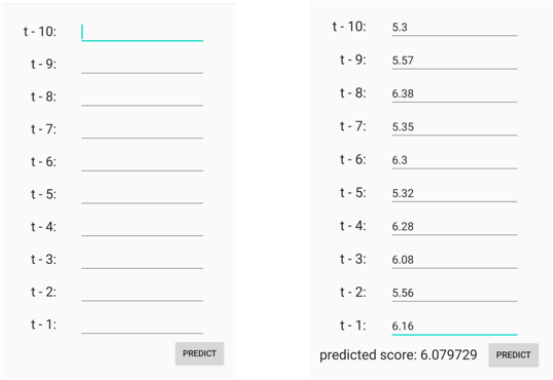

図 1: TF Lite モデルを組み込んだ時系列データ予測アプリケーション

4.2 予測精度

サーバと端末の実行環境の違いによる予測精度の変化の有無を確認するため, 両環境での TF Lite モデルによる予測結果を比較した (表 1). サーバ上と端末上での予測値の差は最大でも 10^{-6} 程度で, ほぼ変わらない精度で予測が可能であることがわかった. なお, TF モデルと TF Lite モデルの予測値を比較したところ, モデル変換により予測が大きく変わることはないことが確認されている.

4.3 予測時間

続いて検証用アプリで予測を行う際にかかる時間を計測したところ, 約 16.8 ミリ秒という結果になった. サーバでの予測時間は約 7.6 ミリ秒であり, 端末上での予測は 2 倍以上の時間を要しているが, 今回想定する輻輳制御 [2] によると制御の周期が 0.3 秒程度であるため, この速度差は許容範囲内であるといえる.

サーバ上での予測値	端末上での予測値
1.8770345449	1.8770333529
0.3401488066	0.3401483297
5.2305030823	5.2305030823
6.3132376671	6.3132376671
1.3176695108	1.3176695108
0.0441148952	0.0441144779

表 1: サーバ上と端末上での予測結果の比較 (抜粋)

4.4 CPU 使用率

最後に, 検証用アプリ使用時の Android 端末の CPU 使用率を計測した. 検証用アプリにおける CPU 使用率は予測処理時において 5%であった. TensorFlow Lite 公式サイトが紹介するサンプルの機械学習アプリで同様の検証を行ったところ, 予測処理時における CPU 使用率は 15%という結果となり, この結果と比較しても十分低い値であるといえる.

5 まとめと今後の課題

本稿では, 端末上予測処理の実現可能性を検証すべく, Android 端末上でトラフィックの輻輳を予測し制御を行うシステムを想定し, サーバで学習させた深層学習モデルを検証用アプリに組み込みその性能を評価した. 予測精度, 処理速度ともにサーバには劣るものの十分有用なものであることがわかり, 端末上時系列データ予測処理の実現可能性が示された.

今後の課題としては, 想定する輻輳制御システムの状況により近い形での実装を進め, その性能を再度評価していきたいと考えている.

参考文献

[1] TensorFlow Lite, https://www.tensorflow.org/lite?hl=ja.

[2] Ai Hayakawa, Saneyasu Yamaguchi, and Masato Oguchi, Suggestion and Implementation of the Cooperative Congestion Control Mechanism for Reducing the TCP ACK Packet Backlog at the WLAN Access Point, Proc.DEIM2015, C2-2, March 2015.

[3] Yamamoto A. et al.(2019), Prediction of Traffic Congestion on Wired and Wireless Networks Using RNN, Proceedings of the 13th International Conference on Ubiquitous Information Management and Communication (IMCOM) 2019. IMCOM 2019. Advances in Intelligent Systems and Computing, vol 935. Springer, Cham.