

SFC制御自動化のための機械学習モデル および再学習フレームワークの研究

<u>平山孝弘</u>地引昌弘 ベドカフレ 情報通信研究機構(NICT)

*本研究の一部は、総務省委託研究「革新的AIネットワーク統合基盤 技術の研究開発(JPMI00316)」による支援を受けて実施している。

Takahiro Hirayama [<u>hirayama@nict.go.jp</u>] ©National Institute of Information and Communication Technology

1



発表概要

- ・ 背景-ネットワーク仮想化(NFV)/サービス機能チェイン(SFC)基盤の制御の自動化
- サービス機能チェイン(SFC)基盤制御への機械学習活用
 - SFC資源割当自動制御の概要
 - 計算機資源自動調停のためのリソース需要予測と再学習
 - アンサンブル学習による予測と二段階再学習
 - SFCマイグレーション(移行)スケジューリングのための機械学習モデル
 - Encoder-Decoder Recurrent Neural Network (ED-RNN)
 - 実機検証に向けた環境構築



5G時代のスライシング





背景 ~ NFV and SFC ~

- Network Function Virtualization (NFV) :
 - 複数のネットワーク機能(NFs)を汎用サーバからなる基盤上でソフトで実装
- Service Function Chaining (SFC) :
 - 必要な複数のNFsを分散配置し、特定の経路に沿って転送されるパケットに対して適切な順番でネット ワーク内処理を施すための仮想的なサービス機能チェインを構築
 →NFsを柔軟に分散配置でき、エンドユーザ毎に適時適切なサービスを迅速に提供可能
 - •参考: IETF RFC 7665 / RFC 8300, ETSI GS NFV-EVE 005 V1.1.1



*資源:コンピュータ資源(CPU、メモリ、ストレージ容量、ソフトウェア基盤)



背景 ~求められる自動化~







サービスネットワーク基盤制御 全体的な課題

- 資源利用状況変動や障害発生時にも、5G時代における多種多様なサービス品質要求を満たすことが必要
 - 既存手法(例:手動操作や最適化手法等)では、処理に時間がかかり、環境変動に追従できない

- 各サービスを提供するための計算資源が不足し、サービス品質が劣化

 サービス品質の維持と有限な計算資源の有効利用を図りつつ、環境変動に追従する高速な計算資 源自動制御機構を備えた、サービス機能チェイニング基盤(データ・サービス・制御・管理 各々 プレーン)が必要



機械学習活用を含め、AIサービスネットワーク 基盤制御技術の研究開発の取組みを紹介



発表概要

- 背景-ネットワーク仮想化(NFV)/サービス機能チェイン(SFC)基盤の制御の自動化
- ・ SFC基盤制御への機械学習活用
 - SFC資源割当自動制御の概要
 - 計算機資源自動調停のためのリソース需要予測と再学習
 - アンサンブル学習による予測と二段階再学習
 - SFCマイグレーション(移行)スケジューリングのための機械学習モデル
 - Encoder-Decoder Recurrent Neural Network (ED-RNN)
 - 実機検証に向けた環境構築





発表概要

- 背景-ネットワーク仮想化(NFV)/サービス機能チェイン(SFC)基盤の制御の自動化
- ・ SFC基盤制御への機械学習活用
 - SFC資源割当自動制御の概要
 - 計算機資源自動調停のためのリソース需要予測と再学習
 - アンサンブル学習による予測と二段階再学習
 - SFCマイグレーション(移行)スケジューリングのための機械学習モデル
 - Encoder-Decoder Recurrent Neural Network (ED-RNN)
 - 実機検証に向けた環境構築



想定される課題と解決策

サービス機能を、応答時間の増加や途絶なく、過不足なく提供することが必要。 ⇒計算資源利用状況変動に合わせてリソース割当量を動的に制御して有効利用





サーバ内VNF間資源調停のためのトラヒック予測

- スタッキングのような強/弱の2段階構成の予測フレームワーク
 - コントローラが集約したトラヒック情報に基づき弱学習器群をトレーニング
 - サーバは弱学習器群と強学習器を用いてトラヒックを予測
 - 強学習器はサーバに到着するトラヒック情報でトレーニング
- 弱学習器群と強学習器をそれぞれ異なるデータサイズ/周期で再学習 •
 - 弱学習器群:多量のデータ。長周期(数週間に1回)の再学習(Forgetting)
 - 強学習器: 少量のデータ。短周期(予測精度悪化時、数時~数日に1回)の再学習 (Dynamic Ensemble)



2. サーバでの学習および予測



トラヒック予測方法

• 時刻[t_1, t_2]のトラヒックから、時刻[t_3, t_4]の平均トラヒックを予測 - $t_2 \ge t_3$ の間はマイグレーションなどの処理のためのマージン

> t_1 (=10:50) $t_2(=11:49)$ $t_3(=12:00) t_4(=12:19)$ 3000 Traffic arrival every 1 min. Interval Traffic arrivals 0007 0007 2000 (I(=10) min.)Avg. traffic every 20 min. Monitoring Window **Prediction Objective** (M(=60) min.)(Avg. value within 20 min.) 0 11:00 11:30 12:00 12:30 13:00 Time



トラヒックデータの(再)学習

• 2018/1/30~2018/2/28までの30日分のWebサーバリクエスト数を学習

- 日中(12:00~19:59)、夜間(18:00~25:59)、および両方(12:00~25:59)の3データセットを用意





再学習による予測精度の向上

- 2018/3/1~4/29までの60日間のトラヒックを予測し性能を評価
- (左図)再学習ありは再学習なしに比べると予測精度が大きく向上
 - 実トラヒック(破線)に近い推移を描く
- (右図)再学習あり(DEのみ、DE+FG)は他の予測手法に比べて過剰/過小な資源割当発生率を低減 (ARMAやRNNのほぼ半分)





発表概要

- 背景-ネットワーク仮想化(NFV)/サービス機能チェイン(SFC)基盤の制御の自動化
- ・ SFC基盤制御への機械学習活用
 - SFC資源割当自動制御の概要
 - 計算機資源自動調停のためのリソース需要予測と再学習
 - アンサンブル学習による予測と二段階再学習
 - SFCマイグレーション(移行)スケジューリングのための機械学習モデル
 - Encoder-Decoder Recurrent Neural Network (ED-RNN)
 - 実機検証に向けた環境構築





AIによるSF移行スケジューリング(目標)

- サーバの資源量、SFチェインの需要変動、VNFのリソース要求量などを参照して、VNFの移行先 &移行時期を<u>すばやく(数秒程度で)</u>決定
 - 数ステップ先を見据えて、同じVNFの移行が頻繁に起きないように配慮





SFチェイン移行スケジューリング問題の簡略化

- スケジューリング問題を下記のように簡略化して、AIのモデルの選定、チューニング
 - 各SFチェインを構成するVNFの中の一つに着目
 - リソース要求はCPUのみ、VNF間の依存関係も考慮せず
 - 時刻ごとに各SFチェインの要求するリソース量が変動する中で、複数あるパスのどのパスにどのSFチェ インを通すか?





移行スケジューリングAI作成の方針

- エンコーダ・デコーダ型のRNN (ED-RNN)を使用
 - 自然言語翻訳等への利用を想定したRNN構成
 - 文章のような、時系列データの処理に有効
 - 入力データを二度読み込み、翻訳精度を向上
 - 一度目の読み込みで入力の前後関係や重要度を試算、二度目の読み込みで解を出力
- 最適化問題の解を教師データとして使用
 - 入力: SFチェインの需要変動の時系列データ
 - 出力: 資源不足・VNF移行回数を最小にする移行スケジュール



Cf. 1. 平山、地引、カフレ IN研究会(2019年3月) 2. Hirayama, Jibiki, Kafle, IEEE Netsoft (2019年7月)

(RNN: リカレントニューラルネットワーク)



教師データの作成

 時刻ごとのサーバの資源不足量S_i(t)と、移行回数m_{i,f}(t)の総和を 最小化するVNF移行スケジュールを定式化して解を探索 i: サーバ、t: 時刻、f: VNF
 - x_{i,f}(t)(∈ 0,1)の最適な組み合わせを出力(tにiがfを収容→x_{i,f}(t) = 1)





エンコーダ・デコーダRNN(ED-RNN)の構成と仕組み

- 時刻 t_1 から t_T (T = 10)までの入力をEncoder, Decoderで2度読み込む
 - RNNを構成するセルにはGRU(Gated Recurrent Unit)を使用
- Encoderの最終セル (t_T) の隠れ層出力 $(h_e(t_T))$ に加え、 各時刻の出力(Attention)を記憶し、Decoderへ渡す
 - Attentionを導入し、時刻 t_1 から t_T の中で重要な時刻がいつかを決定する
- Decoderは前時刻の出力を参照し次状態を決定するので、 前時刻との差分(移行回数)を低くすることが可能になる





評価結果(1/2)

図(a): ED-RNN(Attentionあり/なし)のどちらも同程度の学習効率 - 一方、DNNは学習回数を増やしても損失関数はほとんど変化せず

図(b): ED-RNN(Attentionあり)とランダムは資源不足量を同程度に軽減

- 最適解のおよそ3倍だが、DNNやAttentionなしに比べると非常によい



22



評価結果(2/2)

図(c): ED-RNN(Attentionあり)はVNF移行回数を少なく抑えられている - 一方、ランダムは資源不足量は小さいが移行回数が非常に多い

図(d): (b),(c)の結果から、目的関数の値もED-RNN(Attentionあり)が次善 – 最適解にはおよばないが、ランダム、DNN, Attentionなしより良い



23



発表概要

- 背景-ネットワーク仮想化(NFV)/サービス機能チェイン(SFC)基盤の制御の自動化
- ・ SFC基盤制御への機械学習活用
 - SFC資源割当自動制御の概要
 - 計算機資源自動調停のためのリソース需要予測と再学習
 - アンサンブル学習による予測と二段階再学習
 - SFCマイグレーション(移行)スケジューリングのための機械学習モデル
 - Encoder-Decoder Recurrent Neural Network (ED-RNN)
 - 実機検証に向けた環境構築



実機検証に向けたAI実装





サービス間CPU割当量自動調停の簡易評価

- サービス利用量(VNFに到着するトラヒック量)を予測し、 CPUコア数を自動で増減
- サービスに優先度(高or低)を設定し、 高優先サービスのCPUリソースを確保
 - 高優先(QoS保証型)サービスのリソース確保のための自動調停
 - 両用可能なCPUコアのうち、
 何コアを高優先サービスに割り当てるかを制御
 - 今回は下図における境界①or②を予測に基づき選択
 - 高優先サービスを最大2でつ収容できるように空きCPUを予約
 - 低優先(ベストエフォート)サービスはCPUを共有







まとめと今後の課題

- 周期の異なる2段階の再学習による需要予測と自動資源調停
 - アンサンブル学習で用いる弱学習器群の定期的(長周期的)な再学習(Forgetting)
 - 長期的なトレンドの変化に追随
 - 強学習器の性能劣化時に随時(しきい値ベース)行う再学習 (Dynamic Ensemble)

- 再学習により予測精度が向上し、資源の有効利用を実現

- エンコーダ・デコーダRNNを用いたSFC移行スケジューリング
 - 時系列データをエンコーダ、デコーダで2度読み込み解析
 - 最適化問題の解を、教師データとして学習させる
 - リソースの不足を回避しつつ、VNFの移行回数も抑える
- 今後の課題
 - 数十~百ノード規模での実機検証(JOSE利用)
 - 他社との連携実験



(補足1) Attn-GRUの概要

- 入力 $I(t_k)$ に、前段の出力 $O(t_{k-1})$ とAttentionを組み合わせて入力
 - Linearセル(Y=wX+b)やMultiplyセル(行列積)などを使用
 - 出力はone-hotベクトル形式
 - 一番高い値を示した要素のインデックスがVNF-fを収容すべきサーバ
 - Ex. O_f = (0, 0.3, 0.7, 0.4)と出力された場合は、VNF-fはサーバ2に置くのが適切
 - ちなみに、教師データは $O_f = (0, 0, 0.1, 0)$





(補足2)学習および性能評価シナリオ

- 15,000パターンの教師データを作成し、100サイクル学習
 - サーバ数 $N = 3 \sim 5$, SFC数 $F = 5 \sim 15(\neg \lor \neg \lor \bot)$ 、 $T = 10, P_s = 1, P_m = 0.01$
 - e(t), h_e(t), h_d(t)の大きさ(H)はそれぞれ200、ソルバはAmsgrad(学習率0.001)
 - 15,000個のデータの入力順は毎回ランダム
- 1,000パターンの検証データ(教師データとは)を用いて性能評価
 - 評価指標
 - 資源不足量(平均値)
 - VNF移行回数(平均值)
 - 目的関数(資源不足量+0.01*VNF移行回数)
 - 比較対象
 - 最適解(教師データと同じ解法)
 - ランダム (毎時VNFをランダムに振り分け)
 - ED-RNN (Attentionなし)
 - $3 \mathbb{P}(L = 3) \mathcal{O} \text{DNN}(l_1, l_2, l_3) = (500, 600, 400)$
 - 時刻 t_1 から t_T の状態を一つのテンソルにまとめて入出力

