

実世界に存在する物体あるいは環境

- Virtual Entity

実世界に存在する物体や環境を仮想的に表したもの

2.3 Tensorflow[1]

Tensorflow はデータフロー グラフを使用して数値計算を行なうオープンソースソフトウェア ライブラリである。データの読み込みや前処理, 計算, 出力等の処理を一貫してデータフローグラフを構築して取り扱うアーキテクチャに基づいて実現されている。

2.4 デザインパターン

ソフトウェア設計時の典型的な問題に対し, その解決策に名前を付けて整理し, 再利用しやすいようにカタログ化したものである。我々の設計したアーキテクチャでは Facade パターンと Command パターンを適用している。

2.4.1 Facade パターン

複雑なサブシステムを隠蔽し, 1 つのインターフェイスを提供することによりサブシステムの利用を容易にすることを目的として高インターフェイスを定義することができる。1 つのシステムを複数のサブシステムに分割して構築することはシステムの複雑さを軽減するのに役立つ。

構成要素として 2 つのクラスがある。

- Facade クラス

サブシステム内のどのクラスが要求に対して責任を負っているかを知っており, 要求をサブシステム内の適当なオブジェクトへと送る。

- サブシステム内クラス

サブシステムの機能を実装しており, Facade オブジェクトから割り当てられた処理を行なう。サブシステム内クラスは Facade オブジェクトへの参照を保持しない。

2.4.2 Command パターン

コンピュータに対して発せられる複数の要求自体をオブジェクトとして引数に渡すことで, それらを組み合わせて利用することを可能にする。また, 命令をオブジェクトとして管理するので, 命令の履歴管理や取消し機能の実装等が容易である。

構成要素として 5 つのクラスがある。

- Command クラス

オペレーションを実行するためのインターフェイスを宣言する。

- ConcreteCommand クラス

Receiver オブジェクトとアクションの間のつながりを定義し, Execute オペレーションを Receiver オブジェクトに対して当てはまるオペレーションの呼び出しを行なうように実装する。

- Client クラス

ConcreteCommand オブジェクトを生成して, それに対応する Receiver オブジェクトを設定する。

- Invoker クラス

command に要求実行を依頼する。

- Receiver クラス

要求実現のためのオペレーションをどのように実行するのかを知っている。任意のクラスが Receiver になり得る。

3 設計

本研究の提案する参照アーキテクチャの設計指針と, それに従って設計した参照アーキテクチャを示す。

3.1 設計指針

家電製品利用のユーザ体験向上を目的として, 本研究では機械学習を利用して IoT 家電製品を制御するシステムを設計する。

本研究では, 以下の標準のアーキテクチャを参考にすることで既存の製品にも応用できるように参照アーキテクチャを拡張する。

- IoT-A が提案する参照アーキテクチャ

- Tensorflow アーキテクチャ

Tensorflow は現在最も使われている機械学習のソフトウェアライブラリの一つであり, 幅広いプログラミング言語と OS に対応している。IoT-A は無数の IoT 機器にも対応する参照アーキテクチャの提案をしている。

Tensorflow を参考にして, IoT-A のコンポーネントの役割を機械学習の実現するためのものに具象化する。また, 実世界に存在するモノを扱うシステムを設計するので実世界のデータを自然にモデル化しやすく, 再利用に適したオブジェクト指向で設計する。これにより, 具象化するときに必要に応じてモジュールを入れ替えできる設計になる。

本研究ではアーキテクチャ上での研究課題として, 機械学習を構成するアルゴリズムやモデルなどの各コンポーネントを自由に組み替えられる設計にすることがある。したがってシステムを設計するときに以下のデザインパターンを適用する。

- Facade パターン

- Command パターン

サブシステム間のやり取りや依存関係を小さくできるので Facade パターンを適応し, クラスの再利用性を向上できるので Command パターンを適応する。

3.2 機械学習を考慮した IoT-A

IoT-A の参照アーキテクチャに複数のコンポーネントを追加した具象アーキテクチャとなっている。

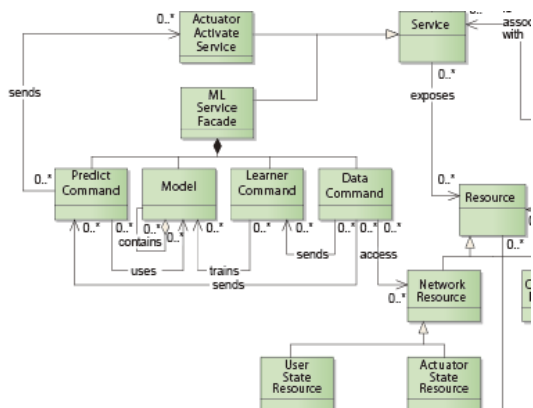


図2 機械学習を考慮した IoT-A の参照アーキテクチャ (一部)

3.2.1 機械学習部分

Tensorflow に基づき手続きとオブジェクトをそれぞれモジュール化して定義することで、これらを独立して変更可能なオブジェクトとする。前処理を Data Command, 最適化アルゴリズムを Learner Command, 推論を Predict Command, モデルを Model とした。

IoT-A におけるサービスはシステムがユーザに対して提供しなければならない機能を保有するものなので、機械学習機能はシステム内においてソフトウェアが提供するサービスとして多相型を定義した。

機械学習は複数のコンポーネントにより実現されるので、サービスに Facade パターンを適用してコンポーネントの関係として定義した。Facade パターンを適用することでシステムは機械学習のサブシステムを意識せずに機械学習をサービスとして利用することができる。

機械学習は、コンポーネント間の次のメッセージ通信により実現される。Data Command は Resource よりデータを読み込んで前処理を行ない、学習部と推論部に引き渡す。Predict Command は前処理されたデータを受け取って予測を行なう。システムは予測を基にアクチュエータを制御する。Learner Command が予測した結果を評価して Model を更新することで学習する。

3.2.2 Resource のサブクラス

Network Resource のサブクラスとして User State Resource と Actuator State Resource を追加した。

機械学習時の訓練データとなるユーザとアクチュエータの履歴を保持しておく必要があると考え、ユーザのバイタルデータとアクチュエータの制御データの保存場所を明確にした。それぞれ User State Resource と Actuator State Resource と定義した。

3.2.3 Service のサブクラス

Service には Actuator Activate Service と ML Service Facade の二つのサブクラスを追加した。

Actuator Active Service は算出された予測を基に制御命令をアクチュエータに伝える。具体的には制御命令を格納している関数を http などのプロトコルに変換することで、アクチュエータが理解できる形式に変換して送信する。Actuator Activate Service は IoT-A の定義に従って Service とした。ML Service Facade は既に解説したので省略する。

4 家電制御システムの設計

本研究で提案した参照アーキテクチャを具象化した設計例を示す。

4.1 設計の具象化

設計例として、エアコンの制御システムを示す。家庭用エアコンはほとんどの家に備え付けられており、室内の環境を最も制御しやすい機器の一つであると考えて採用した。ウェアラブルデバイスはバイタルデータを記録でき、身に着けるうえでユーザが感じるストレスが少ないと考えた。

この具象例を踏まえ、アクチュエータ (Actuator) を家庭用エアコン (Air Conditioner), センサ (Sensor) をウェアラブルデバイス (Wearable Device), 機械学習手法 (ML Service Facade) を LSTM(LSTM Service Facade) と定義した。

LSTM は時系列データを扱えるうえに、長期依存を学習できることから採用した。

図3に具象アーキテクチャを示す。

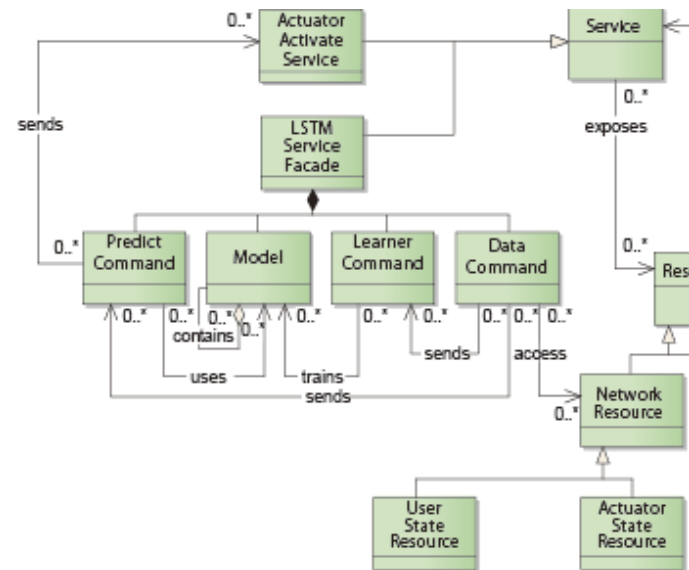


図3 エアコンの制御を考慮したシステム設計例 (一部)

この具象例では LSTM や松本らが提案しているポアンカレプロット [6] などを利用することで実現できる。

我々のシステムは機械学習部分が4つのコンポーネントに分けられているので、この具象例においても LSTM やポアンカレプロットを必要に応じて適応することができた。

下図にシステム設計例の配置図を示す。

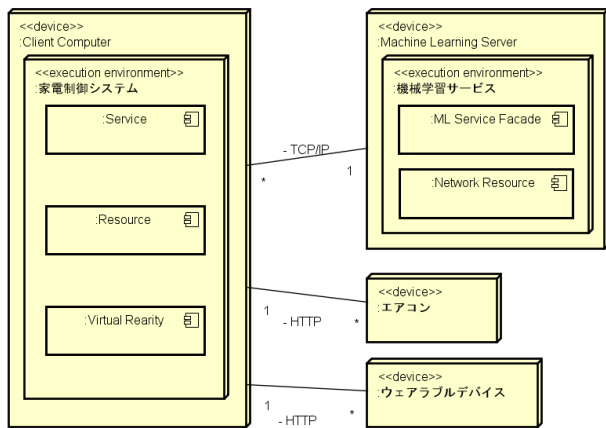


図 4 システム設計例の配置図

本研究では、家電制御システムと機械学習サーバとデバイスをそれぞれ別のノードとした。家庭用コンピュータには機械学習に必要な計算能力を持ち合わせていないので、クラウド上のサーバにサービスを移行した。

5 考察

本研究で提案したシステムと関連研究を比較し、利点及び欠点を考察していく。

5.1 川原らの研究 [4]

川原らの研究は履歴を機械学習して導出した推奨制御をコンピュータに表示し、ユーザに制御を選択させる。

機械学習による予測のほかにルールベースのポリシー処理エンジンを追加することで、稀な事象にも対応できるうえ、安全に配慮した制御をあらかじめ設定することができる。

5.2 Chen らの研究 [2]

Chen らはユーザのバイタルデータを機械学習することで病状を正しく判別し、緊急通知の誤報を減らすシステムを提案している。このようなシステムを設計することで医療従事者をサポートし、医療現場の環境を改善する。

Chen らの研究は、バイタルデータに対して 87% の精度でノイズを正しく判別できる機械学習モデルを示した。

5.3 本研究の位置づけ

本研究ではオブジェクト指向でシステムを設計した。図 4 のように機械学習部分をクラウドに移したことで、家庭向けコンピュータにおける実行効率を向上できると考える。また機械学習は入力データをもとに膨大な演算処理を行なう必要があることから、消費電力が増加するという問題がある。近年機械学習を行なうことを考慮したコンピュータが開発されており消費電力は削減されている。本研究では機械学習をサーバ上で行なうので、今後この問題は改善させると考える。ユーザが増加するとサーバへのアクセスが集中する恐れがあるが、既存の負荷分散技術などを適応できると考える。

● 川原らの研究と比較して

川原らのシステムは推奨された複数の選択肢のうち、どの選択肢がユーザによって選ばれたかに基づいて学習する。我々のシステムは家電製品を制御する前後のバイタルデータを比較して学習する。どちらも家電製品の制御時に学習するという観点からは有効な手法である。

我々のシステムでは自動的に家電製品を制御するので、制御時にユーザの時間と労力を消費しないことからユーザ体験が向上すると考える。

川原らのシステムは、ポリシーを設定することで学習に必要な訓練データが十分でない場合も制御を行なえる。我々のシステムは、稀な事象に関しては十分に訓練データを集めることができない可能性がある。

● Chen らの研究と比較して

我々のシステムは測定時に発生するノイズを考慮しないので、ノイズに基づいた誤った学習をする可能性がある。一方、バイタルデータ自体を機械学習しないので大量の訓練データを必要としない。

6 おわりに

本研究ではユーザ体験の向上を目的とした学習機能を持つ家電制御システムの設計した。システムが能動的に家電製品を制御するのでユーザ体験が向上することが本研究の利点だと考える。実装までは至らなかったため今後は実際に試作品を制作して妥当性を検証する必要があるうえ、川原らの研究や Chen らの研究で提案されている手法を本研究に組み込むことも必要であると考えます。

参考文献

- [1] Abadi, M., Barham, P., Chen, J., et al.: Tensorflow: a system for large-scale machine learning, *OSDI*, Vol. 16(2016), pp. 265-283.
- [2] Chen L., Dubrawski, A., Wang D., et al.: Using Supervised Machine Learning to Classify Real Alerts and Artifact in Online Multisignal Vital Sign Monitoring Data, *Crit Care Med*, 2016.
- [3] Gamma, E., Helm, R., Johnson, R., et al.: *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1999.
- [4] 川原圭博, 司化, 猪鹿倉知広ほか: 行動履歴と制約条件を考慮した情報家電制御機構, 情報処理学会研究報告ユビキタスコンピューティングシステム (UBI), 2006, pp. 55-60.
- [5] Martin, B., Mathieu, B., Nicola, B., et al.: The Internet of Things – Architecture, Ver. 3.0(2013).
- [6] 松本佳昭, 森信彰, 三田尻涼ほか: 心拍揺らぎによる精神的ストレス評価法に関する研究, ライフサポート, Vol. 22, No. 3(2010), pp. 105-111.