

情報通信プロジェクト G 班中間報告書

13EC602 郭柏辰, 14EC004 飯田頌平, 14EC552 陳玉皓, 14EC602 劉宇航

2017 年 7 月 29 日

1 企画の目的

既存の天気予報は都市毎にしか見ないものであったり、雨雲の動きを生データで実況するものが多く、気象予報士でもない一般人が特定の位置の天気を予報するには向いていない。

そのため、センサを活用して局地的な天気を予測できる IoT デバイスに需要があると考えられる。

2 構成

局地的な情報を計測する必要があるため、センサは持ち運びができるものでなくてはならない。

センサが測定したデータを取り出す部品も、同様にコンパクトである必要がある。

そこで、データ取得を行う部品に RaspberryPi を採用する。RaspberryPi は名刺ほどの小型のコンピュータであり、MicroSD カードにシステムやデータを保存する。

RaspberryPi は GPIO 端子を通して I2C(Inter Integrated Circuit) 接続を行えるため、I2C に対応したセンサである BME-280 を用いて、温度・湿度・気圧を計測する。

こうして小型の部品を活用することで、どこにでも天気予報機を持ち運んで天候のデータを測定することができる。

測ったデータから天気予報を行うには、ニューラルネットワーク [1] というアルゴリズムを採用する。

RaspberryPi は Linux ベースの OS で動いているため、Python プログラムで実装したニューラルネットワークを使って予測を行える。

予測結果は、RaspberryPi の GPIO 端子に接続するタイプの小型ディスプレイを通してユーザに伝えられるようにする。

以上のことから、構成部品をまとめたものを下に示す。

表 1 構成部品一覧

種別	名称	値段
マイコン	RaspberryPi3 ModelB	5,600 円 (a)
MicroSD カード	SanDisk Ultra 40MB/s 8GB	798 円 (b)
温湿度・気圧センサ	BME-280 モジュール	1,080 円 (a)
ディスプレイ	AE-AQM0802 モジュール	700 円 (a)
オス-オスジャンパ	10cm 20本セット	180 円 (a)
オス-メスジャンパ	15cm (赤) 10本セット	220 円 (a)
ブレッドボード	BB801	200 円 (a)
	合計	8,778 円

価格は (a) 秋月電子 (b)amazon 調べ

3 内部仕様

3.1 ブロック構成図・回路図

ブロック構成図を 1 に、回路図を図 2 に示す。

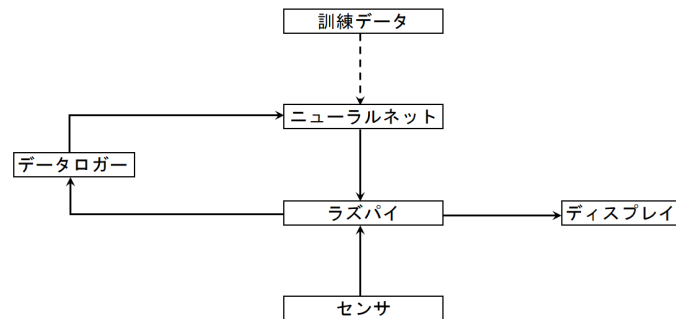


図 1 ブロック構成図

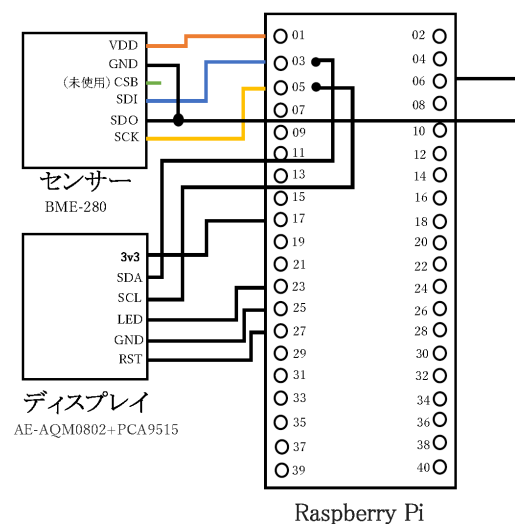


図 2 回路図

RaspberryPi には I2C でセンサとディスプレイが接続されている。センサおよびディスプレイはブレッドボード上に配置される。

3.2 ソフトウェア

3.2.1 OS

RaspberryPi を動かす OS には、Raspbian を採用する。これは RaspberryPi が公式にサポートしている OS であり、Ubuntu ベースで作られているため、bash を用いて通常の Linux のようにソフトウェアを動かせるためだ。

3.2.2 センサ読み取り

センサが取得したデータを読み取るソフトには、センサ・モジュールの販売元である SwitchScience が提供しているプログラムを用いる。読み取りプログラムは Python でラップされており、Python 上の str（文字

列) 形式でデータを取得することができる。さらにこのコードを改変して、読み取ったデータを天気予報用のプログラムに送信するようにする。

また、定期的なデータの取得には `bash` コマンドの `crontab` というものを用いる。これによって、一時間毎にデータを読み取るコマンドを自動的に実行できる。

3.2.3 天気予報

天気予報を行うプログラムとして、Python でニューラルネットワークを実装したコードを用いる。

Python は深層学習用のライブラリを多数有しており、ニューラルネットワークを用いたプログラムの実装に向いているためだ。

実装にあたって、深層学習用ライブラリの Keras を採用する。Keras はバックエンドのライブラリに数値計算の処理を任せており、勾配法の実装などの詳細な点を無視しながら比較的簡単にニューラルネットワークを実装できる。

今回用いるものは LSTM[2] と呼ばれるモデルだが、これは時系列の情報を学習することができ、通常のニューラルネットワークよりも時間の変化による予測値の変化に敏感になるためだ。

一般的に、ニューラルネットワークによる予測には、事前に学習用データを用いてネットワークの訓練を行う必要がある。学習用データには、気象庁が用意している CSV データを採用する。これは不要な情報が多々含まれているため、Python プログラムによって必要な情報だけを抽出する。

抽出する情報は、一日あたりの平均温度・平均湿度・平均気圧・降水量の4項目で、これらを一年分用意する。降水量についてはしきい値を取り、一定量の降水量があったときに雨天であると判定し、ブール値でフラグを立てるように変換する。

そうして生成されたデータを逐次的に LSTM に入力し、誤差逆伝播法 [3] のアルゴリズムによって天気予報に適したモデルを生成する。天気予報モデルは、センサから得た温度・湿度・気圧の3次元のデータに基いて、未来の降水確率を出力する。

なお、現段階では RaspberryPi に組み込んで使用するため、マシンパワーに不安がある。そのため、簡単なネットワークを組むことにした。具体的には、以下の構成からなるニューラルネットである。

- LSTM 層, 隠れユニット数:10
- 全結合層, ユニット数:10
- 活性化関数, シグモイド
- 誤差関数, 二乗誤差

最適化手法は勾配降下法的一种である RMSProp[4] を採用した。

また、活性化関数については、LSTM 内の各ユニット間にシグモイド関数とハイパボリック・タンジェント関数を用い、全結合層と出力層の間にはシグモイド関数を用いている。出力層の前の非線形関数がシグモイドなのは、出力値を確率値として表現するためである。

3.2.4 ディスプレイ

採用したディスプレイ AE-AQM0802 は、RaspberryPi から `bash` コマンドで動作させることができる。Python の `subprocess` モジュールから `bash` を操作することで、間接的に Python プログラム上からディスプレイの操作ができる。

4 外部仕様

4.1 外観

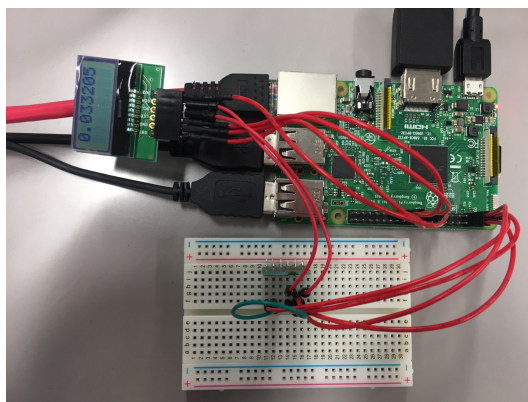


図3 現在の外観

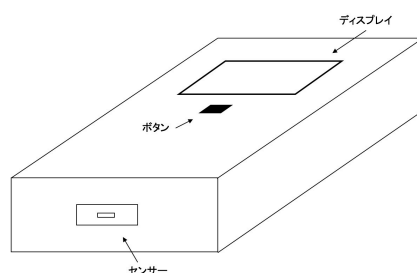


図4 完成予想図

本デバイスの外観を図3に示す。予算に余裕があるならば、さらに図4のようにケースで覆い、ボタンを押すことで予測を開始できるようにする。

4.2 使用方法

本デバイスの使用方法は、以下の手順に則る。

1. デバイスの電源を入れて、起動する
2. デバイスを適当な観測箇所に設置する
3. ディスプレイに出力された天気予報を確認する

センサによる測定を行う上で、屋外での動作が見込まれているが、予算の都合上、雨よけのケースや携帯型のバッテリーを不採用にしたので、実際の運用では窓の近くに設置する形が想定される。

ディスプレイの出力値は、単純に確率を数値で表したものと、その確率をどう判断すべきかの目安となる文字列を表すことにする。これによって、ユーザは傘を持ていくかどうかという、もっとも重要な情報を即座に判断できるようになる。

また、実際に運用する際には、観測地点の温湿度データを記録し続ける必要がある。そのため、据え置き型のデバイスとして使うことになる。初使用時など、十分なデータを記録出来ていない場合には、気象庁が公開している過去の観測データを代わりに使用した予測を行うようにする。

5 作業計画

5.1 現在の進捗状況

当報告書の内部仕様・外部仕様に記載した点は、一通り完了した。具体的には、

- RaspberryPi のセットアップ
- センサーによるデータの取得
- ディスプレイへの表示
- ニューラルネットによる予測

ここまでが一通り完了し、ターミナルからコマンドを打つと、センサから取得した値を用いた予測結果がディスプレイに表示されるようになっている。

なお、現在の予測精度はおおよそ 70% である。気象庁の発表によると、2015 年の降水の有無の的中率は 85% を越える [5] ので、当デバイスの目標精度も 85% とする。

5.2 今後のスケジュール

表 2 スケジュール表

日付	予定
7/31	報告書提出
8/15	モジュールテスト：温湿度ログ管理
8/29	モジュールテスト：クライアント・サーバ間通信
9/12	モジュールテスト：ディープ・ニューラルネットによる予測
9/19	実装テスト、追加目標の定義
9/26	タスク消化
10/3	以後、予備日

表 2 に今後の作業計画を示す。9/19 までは実機に触れないので、それまでは二週間の間隔で各モジュールのテストを行う。各モジュールテストの詳細を以下に示す。

温湿度ログ管理 cron を使ったログの自動取得、保存、読み出しを、センサの値の代わりにダミーデータを用いて確認する。

クライアント・サーバ間通信 研究室にある Linux サーバとノートパソコンの間で、python コード上のソケット通信によるデータのやり取りを確認する。

ディープ・ニューラルネットによる予測 現在は LSTM を一層用いて、一日の平均の温湿度から翌日の降水確率を予測している。これに対し、複数の LSTM と時間単位の温湿度から、数時間後の降水確率を予測できるようにする。

5.3 班内での分担

以下のように班内で作業を分担する。

飯田 ニューラルネット関連プログラム
陳 センサ・ディスプレイ関連プログラム
劉・郭 実機での配線、部品調達、図面作成など

6 総括

中間報告時点で、計画通りに作業が進んでいる。精度や構成などはより改善する余地が残っており、今後はそれらに力を入れる予定だが、既に単独の IoT デバイスとして使用できるだけの完成度に達している。

参考文献

- [1] Schmidhuber, Jrgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015): 85-117.
- [2] Hochreiter, Sepp, and Jrgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
- [3] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., & Jackel, L.D. (1990). Handwritten digit recognition with a back-propagation network. Advances in Neural Information Processing Systems, 2, 396-404, Morgan Kaufman.
- [4] Geoffrey Hinton, Nitish Srivastava, Kevin Swersky. "Neural Network for Machine Learning - Lecture 6a Overview of mini-batch gradient descent"
- [5] 気象庁. "天気予報の精度検証結果" http://www.data.jma.go.jp/fcd/yoho/kensho/yohohyoka_top.html