

## FPGA による熱中症の実時間予測

伊藤 瑞基<sup>†</sup> 山口 佳樹<sup>‡</sup><sup>†</sup>筑波大学情報学群情報科学類<sup>‡</sup>筑波大学システム情報工学研究科

## 1 はじめに

近年、地球温暖化により熱中症で緊急搬送される方が増えている。中でも高齢者の割合が非常に高い。これは、体温調節機能の衰えによって体温が上がりやすく、なおかつ暑さを自覚しにくいいため、熱中症になるリスクが高いからである。熱中症を回避するためには自分の体内状況を確認することが必要である。そこで、リアルタイムで熱中症レベルを検知できるシステムの構築について検討を行う。まず、熱中症のリスクレベルを把握する手法として、自己組織化マップに焦点を当てる。その上で、FPGA ボードを用いたリアルタイム熱中症検知のためのシステムを構築し、実際の熱中症リスクレベルマップを画面に出力することで、体内状況を把握できるようにする。

## 2 FPGA

FPGA(Field Programmable Gate Array)とは、論理関数を書き込むことで利用者が望む機能を実装することができる集積回路である。用途ごとの専用の回路として最適化し、記述することができるため、汎用的な CPU などよりも高速な処理や低い消費電力が期待される。また、小さいものだと 5mm 四方であるため、腕時計のようなウェアラブル端末に実装する際に装着者の負担になりにくいという利点もある。

## 3 SOM マップ

SOM(Self-Organization and Associative Memory)は自己組織化マップとも呼ばれており、フィンランドの研究者 T. Kohonen の発明したニューラルネットの一種である。また、教師なし学習を行う位相保存写像 (topology preserving mapping) の一種でもある。高次元の観測データを低次元に写像することにより、データの解析を行うことができる。特に二次元空間に写像する場合、地図のように可視化することから SOM マップと呼ばれる。SOM のアルゴリズムは競合過程、協調過程、適応過程の 3 過程の繰り返し計算である。

## 3.1 初期化

3 過程の繰り返しを行う前にまず参照ベクトルに初期値  $y_k(0)$  を与える。そして、学習回数を  $t=1$  として競合過程に進む。

## 3.2 競合過程

各観測データ  $x_n(n=1, \dots, N)$  に対して、最も近い参照ベクトルを持つノードを勝者ノード (Best Matching Unit: BMU) とする。すなわち、BMU のノード番号  $k_n^*$  を、すべての  $x_n$  に対して求める。

$$k_n^*(t) = \operatorname{argmin} \|x_n - y_k(t-1)\|^2 \quad (1)$$

## 3.3 協調過程

各勝者ノードが近傍ノードに分配する学習量  $R_{kn}$  を計算する。

$$R_{kn}(t) = h(z_n(t), \xi_k; \sigma(t)) \quad (2)$$

$h(z, \xi)$  は近傍関数と呼ばれており、BMU に近いほど多くの学習量の配分を受ける。近傍関数としては、ガウス関数を用いるのが一般的である。

$$h(z, \xi; \sigma) = \exp\left[-\frac{1}{2\sigma^2} \|z - \xi\|^2\right] \quad (3)$$

## 3.4 適応過程

全参照ベクトルを観測データの重み付き平均になるように更新する。

$$y_k(t) = \frac{\sum_{n=1}^N R_{kn}(t)x_n}{\sum_{n=1}^N R_{kn}(t)} \quad (4)$$

そして、 $t=t+1$  として競合過程に戻る。これを学習が収束するまで行う。最終的に得られた  $z_n$  がマップ上でのデータの位置となる

## 4 検討手法

## 4.1 被験者の体内状況の観測方法

まず被験者の体内状況 (心拍数や呼吸数) のデータを取る。このデータを取るために、GARMIN 社が開発した、ヘルスマonitoring 機能搭載のスマートウォッチである vivoactive4 を用いた。この vivoactive4 は光学式の心拍計を搭載しており、腕につけるだけで装着者の心拍数を計測することができる。

## 4.2 SOM のハードウェア化

本研究では、SOM マップをリアルタイムで生成するために、SOM のアーキテクチャを FPGA に実装可能なサイズで回路構成できるか、また実用的な速度で演算をすることができるかを検証する。しかしハードウェア化にあたり、いくつかの問題が生じた。

## 4.2.1 マンハッタン距離の採用

一つ目はユークリッド距離の計算が極めて大きいことである。SOM のアルゴリズムでは、ユークリッドの距離で BMU を決める。しかし、それでは回路規模が大きくなってしまったため、解決策としてマンハッタン距離 (L1-距離) を用いた。マンハッタン距離は以下のような数式で求められる。

$$d(x, y) = \sum_{k=1}^n |x_k - y_k| \quad (5)$$

## 4.2.2 矩形関数の採用

二つ目はガウス関数がハードウェアで実装することが困難という問題である。SOM のアルゴリズムでは近傍関数にガウス関数を用いるのが好ましいとされている。しかし、回路規模を考え、以下の矩形関数を用いることにした

$$h(z, \xi; \sigma) = \begin{cases} 1 & z \leq \sigma \\ 0 & z > \sigma \end{cases} \quad (6)$$

4.3 実装環境

表1に本研究の実装環境を示す。

表1:実装環境

|           |                      |
|-----------|----------------------|
| FPGA      | Digilent 社 Atlys ボード |
| 開発ソフトウェア  | ISE Design Suite     |
| ヘルスマニタリング | GARMIN 社 vivoactive4 |

4.4 実装概要

システムの概要を図2に示す。

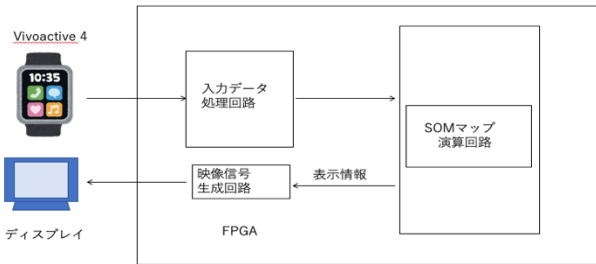


図1: システム概要

5 結果(ソフトウェアでの実装)

ハードウェアで実装する前に SOM マップが実際にどのようなデータマップ化を行うか確認する為、C言語によるシミュレーションを行う

5.1 使用したデータ

C言語でシミュレーションを行うにあたり、国土交通省のホームページにある過去の気象データを用いる。今回はつくば市の2000年1月から2019年11月までの月ごとの最高気温、平均気温、日平均 25℃以上日数のデータを使用する。表2に使用したデータを示す。

表2: つくば市の気象データの一部抜粋

| 年月     | 最高気温(°C) | 平均気温(°C) | 日平均気温 25℃以上日数(日) |
|--------|----------|----------|------------------|
| Jan-00 | 16.1     | 4.8      | 0                |
| Feb-00 | 14.1     | 2.9      | 0                |
| Mar-00 | 19.6     | 7        | 0                |
| Apr-00 | 24.5     | 12.6     | 0                |

5.2 実験結果

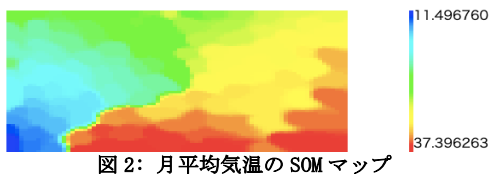


図2: 月平均気温の SOM マップ

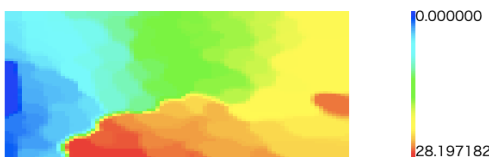


図3: 月最高気温の SOM マップ

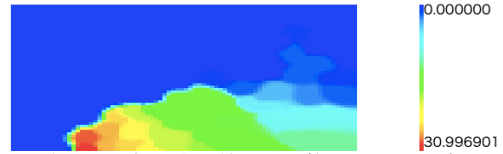


図4: 25℃以上が続いた日数の SOM マップ

5.3 考察

図2, 3, 4を見ると、値が大きくなるほど青から赤に色が変化している。また値が近いものはマップ上の位相関係も近くなるように配置できている為、ソフトウェアでの実装はできていると考えられる。

6 結果(ハードウェアでの実装)

6.1 使用したデータ

表3に使用した心拍数データの一部抜粋を示す。今回はデータの収集が間に合わなかったため、心拍数のみで実験をする。呼吸数や発汗量といったデータの収集も今後の課題としたい。

表3: 使用したデータの一部抜粋

|            |       |              |
|------------|-------|--------------|
| 2019/11/15 | 23:58 | 82 count/min |
| 2019/11/15 | 23:56 | 79 count/min |
| 2019/11/15 | 23:54 | 72 count/min |
| 2019/11/15 | 23:50 | 69 count/min |

6.2 実験結果

図5に心拍数データから SOM マップを生成した結果を示す。青い部分が心拍数の低い領域で、緑→黄色→赤の順に心拍数が高くなっていく。



図5: 実験結果

6.3 考察

実験結果の SOM マップを見ると、水色の領域が多くなってしまい分析が難しくなりました。これは、今回使用した FPGA が小規模であったため、使用できたデータサイズが行 256bit、列 16bit、という少ない抽出量であった為、実験結果が偏ってしまったのだと考えられる。しかし、値の近いものはマップ上でも近い場所に配置することができている為、今後大きな FPGA で実装することができれば、詳細な分析ができると考えられる。

7 終わりに

今後の研究では、心拍数に加え、呼吸数、発汗量も分析できるようにし、FPGA による熱中症の実時間予測を目指す。また回路の最適化も同時に行っていき、手首などに装着できるデバイスを開発できるよう研究を進めていく。

参考文献

[1] Teuvo Kohonen, “自己組織化マップ”, 丸善出版  
 [2] 天野英晴, “FPGA の原理と構成”, オーム社  
 [3] 金子宗治, “高階自己組織化マップのハードウェア化”