

# 推定された習熟度の加工技術教育への適用

福留浩太\*<sup>1</sup>・熊澤典良\*<sup>1</sup>・奈良大作\*<sup>1</sup>・上谷俊平\*<sup>1</sup>・近藤英二\*<sup>1</sup>

Email: kumazawa@mech.kagoshima-u.ac.jp

\*1: 鹿児島大学大学院理工学研究科

◎Key Words IoT, 実習工場, 機械学習

## 1. はじめに

本学大学院理工学研究科には「地域コトづくりセンター中央実験工場（以下、実習工場とする）」と呼ばれる実験・実習工場が設置されている。実習工場ではフライス盤や旋盤、ボール盤などの工作機械を保有し、管理するとともに学生の加工技術教育を行っている。工作機械は高速で回転する切削工具に材料をあてて加工を行うため、不適切な操作を行うと重大な事故につながる恐れがある。重大な事故を防ぐために実習工場では安全教育や職員による巡視を実施している。職員による実習工場内の巡視は作業内容や作業方法に目を配り事故の予防を可能とするが、職員の数は限られており、すべての利用者に対して常に目を配ることは困難である。

先行研究<sup>(1)-(3)</sup>では新たな事故の予防と安全対策を目的として実習工場の IoT 化を進められた。実習工場内の工作機械はネットワークに接続され、利用者や稼働状況の情報はサーバに送信される。稼働状況の情報は実習工場の配電盤内に設置されたクランプ式電流センサより消費電力の傾向を示す電力指標として取得される。電力指標はニューラルネットワークによって解析され、工作機械の操作に対する習熟度は推定される。リアルタイムで推定される習熟度は事故予防や加工技術教育に応用される。

本報告では習熟度の推定に TensorFlow を導入する TensorFlow は機械学習のライブラリとして多く用いられており、NCS2 との親和性も高い。これまで習熟度の学習や推定のために使用していた自作の演算プログラムを TensorFlow ライブラリに変更し、市販の推論エンジンを用いて習熟度を推定する。

## 2. Neural Compute Stick 2 による推定処理

機械加工における習熟度を推定するために本研究では、データの学習に TensorFlow<sup>(4)</sup> ライブラリを用い、推定には図 1 に示される USB 推論エンジンである Neural Compute Stick 2<sup>(5)</sup>（以下、NCS2 とする）を導入した。NCS2 は OpenVINO ツールキット<sup>(6)</sup> で動作し、機械学習の外部演算装置として利用される。NCS2 は予め用意されている多数の学習済みモデルを用いることで高度な顔認識や物体認識を容易に実装できる。もちろん、TensorFlow や Caffe などの機械学習向けのライブラリで独自に作成した学習モデルを OpenVINO ツールキットのモデル・オブティマイザで中間表現（bin ファイルと xml ファイル）にすることで、NCS2 の推論処理に利用できる。

これまでの習熟度の推定は入力層や中間層、出力層の数や層の数に対応した演算プログラムの作成が必要であったが、TensorFlow を用いると入力層や中間層の計算グ

ラフを定義するだけで自動的に演算が行われて、学習が実行される。TensorFlow による学習によって、学習結果である重みとバイアスおよび用いたニューラルネットワークの構造は保存される。

先行研究で用いたニューラルネットワークを TensorFlow で構築し、NCS2 を用いて習熟度の推定を行う手順は以下の通りである。

**Step 1)** TensorFlow を使って、先行研究と同じモデル（入力層等）を作成する。

**Step 2)** 学習を実行する。

**Step 3)** 学習済みモデルを pb ファイルに保存する。

**Step 4)** pb ファイルから、bin ファイル（学習後の各種パラメータ）および xml ファイル（ニューラルネットワークの構造）を生成する。

**Step 5)** bin ファイルおよび xml ファイルを NCS2 に読み込む。

**Step 6)** NCS2 による推論処理を実行する。

これまで我々が使用していたプログラムでは、Step 1 に対応するモデルは行列を用いた数値演算処理によって作成されており、モデルの変更によるプログラムの修正には正確さが要求されていた。本研究で使用したニューラルネットワークの構造は先行研究と同じ入力層のノード数 600、隠れ層のノード数 100 のニューラルネットワークであり、出力層は「習熟度が高い、習熟度が低い、習熟度が不明、電源が切られている」の 4 つの状態である。入力層と隠れ層の結合にはバイアスを与え活性化関数にシグモイド関数、出力層の活性化関数にソフトマックス関数、損失関数として交差エントロピー誤差を用い、バッチ数を 50、学習率を 0.01 とした。ニューラルネットワークの学習には 4 つの状態に対応するデータをそれぞれ 100 組、合計 400 組用いた。

図 2 は NCS2 を使用したリアルタイムに出力される習熟度の推定結果の例を示している。10 分間の電力指標の時系列データに対して、NCS2 による習熟度の推定は毎秒実行され、出力は配列と文字列で表示される。配列の要素である 4 つの実数は出力層に設定した 4 つの状態「習熟度高、習熟度低、習熟度不明、電源切」にそれぞれ対応するソフトマックス関数の出力する確率である。4 つの出力値のうち最も高い確率に対応する状態が推定結果の文字列として表示される。



図 1 Neural Compute Stick 2

```

[[ 0.13671875  0.44287109  0.10461426  0.31616211]]
習熟度低
[[ 0.28759766  0.29321289  0.15844727  0.26074219]]
習熟度低
[[ 0.2364502   0.38378906  0.15161133  0.22839355]]
習熟度低
[[ 0.2746582   0.29736328  0.20239258  0.22558594]]
習熟度低
[[ 0.18286133  0.25708008  0.12078857  0.43945312]]
電源切
[[ 0.09185791  0.1998291   0.11102295  0.59716797]]
電源切

```

図 2 NCS2 を使用したリアルタイム習熟度推定

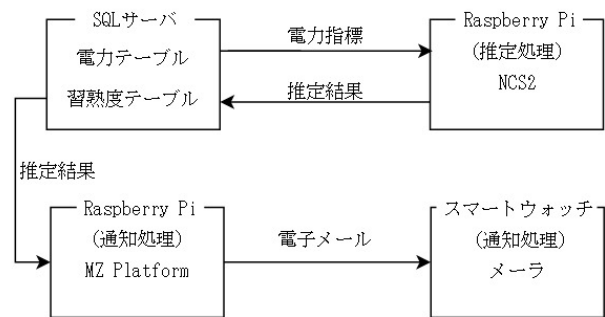


図 3 構成したシステム



図 4 スマートウォッチ

### 3. システムの概要

構築したシステムを図 3 に示す。推定処理部は Raspberry Pi 3 内に組み込まれた Python プログラムと NCS2 によって構成され、MySQL データベースから取得した 10 分間の電力指標に対して NCS2 で推論処理されて習熟度の推定結果を得る。習熟度推定の結果は MySQL データベースの習熟度テーブルを更新する。習熟度テーブルは実習工場内の工作機械名と習熟度からなり、習熟度は推定の結果によって更新される。新たに追加した習熟度テーブルを参照することで他のアプリケーションからも習熟度の推定結果を利用することができる。

通知処理には設計・製造業務を支援する IT システムの開発にたけている産総研独自のソフトウェア開発環境の MZ Platform<sup>(7)</sup>を用いる。MZ Platform で作成されたアプリケーションから毎秒習熟度テーブルを参照し、習熟度フィールドが「低い」場合に MZ Platform のアプリケーションにより職員に通知する。職員に習熟度の推定結果を通知するためのウェアラブルデバイスとして図 4 のスマートウォッチを利用する。スマートウォッチは実習工場内の無線 LAN に接続できるものを選定し、推定結果の通知には電子メールを利用することとした。

スマートウォッチは、実習工場内に設置された無線ルータを介して電子メールを受信する。スマートウォッチで受信されるメールには習熟度の推定結果、機械番号、利用者情報が記載されており、その内容が画面に表示されて職員は対象の作業者を認識できる。職員は直ちに作業者に駆け寄って工作機械の操作手順や加工方法のアドバイスなどの支援を行うことができる。支援を行うことで操作方法の誤りや作業者の焦りから生じる重大な事故を防ぐことができる。

### 4. おわりに

本研究では、機械加工における習熟度の推定に TensorFlow と NCS2 を導入した。TensorFlow によってニューラルネットワークの構造変更は容易となり、得られた学習結果を読み込んだ NCS2 によって習熟度の推定は実行される。NCS2 によってリアルタイムで得られる習熟度推定の結果はスマートウォッチにメールで受信され職員に通知される。通知を受けた職員は対象となる作業員に対して適切なタイミングでアドバイスを行える。今後、アドバイスをを受けた作業員に対してアンケート調査を行い、我々の開発したシステムの効果を検討する。

#### 参考文献

- (1) 熊澤典良, 奈良大作, 近藤英二: “実験・実習工場における教育研究施設としての IoT 化への取り組み”, CIEC 2017 PC カンファレンス論文集, pp.197-198 (2017)
- (2) 野口就平, 熊澤典良, 奈良大作, 近藤英二: “実験・実習工場の IoT 化 -安全対策に寄与するアプリケーションの開発-”, CIEC 九州支部 2017 九州 PC カンファレンス論文集, pp.15-16 (2017)
- (3) 野口就平, 熊澤典良, 奈良大作, 上谷俊平, 近藤英二: “IoT 実証ラボに指定された実験・実習工場における学生への教育効果”, CIEC 2018 PC カンファレンス論文集, pp.223-224 (2018)
- (4) TensorFlow: “TensorFlow Core”, <https://www.tensorflow.org/tutorials/keras?hl=ja>, 2019 年 6 月 11 日アクセス
- (5) Intel: “Intel® Neural Compute Stick”, <https://software.intel.com/en-us/neural-compute-stick>, 2019 年 6 月 11 日 アクセス
- (6) Intel: “Intel® Distribution of OpenVINO™ Toolkit”, <https://software.intel.com/en-us/openvino-toolkit>, 2019 年 6 月 11 日 アクセス
- (7) 国立研究開発法人 産業総合研究所 製造技術研究部門: “MZ プラットフォームユーザー会”, <https://ssl.mo-0987.tresazukuri.org/mzplatform>, 2019 年 6 月 11 日アクセス