

放電加工機の異常検知（深層学習）

1. はじめに

今回は、深層学習(deep learning)を用いた異常検知に取り組みます。

深層学習というと、画像認識が想像されますが、NN(neural network) という概念で紹介されます。ニューラル(neural)とは、人間の脳の神経系の意味で、「ネコ」を見たとして、視神経で、ザックリとした事象として認識し、何層かの神経系を介して、「猫」と認識する過程を、プログラムにして実装するのが、NN(neural network) と言えます。（実際はこんな単純ではないようですが）

深層学習で時系列データをあつかう場合、RNN (recurrent neural network) という概念を用います。recurrent とは 回帰型と訳され、ある時点の入力が、それ以降の出力に影響を及ぼすというモデルになります。深層学習を扱うライブラリとして、TensorFlow+Keras と PyTorch がありますが、今回は、TensorFlow+Keras で行っています。

参考にした 文献、WEB

・時系列解析 -自己回帰モデル・状態空間モデル・異常検知- 島田直希 著 共立出版（株）

・3分で入門！時系列データ分析でよく使うLSTMとは？

<https://aizine.ai/glossary-lstm/>

・LSTM (Long Short-Term Memory)

<https://cvml-expertguide.net/terms/dl/rnn/lstm/>

・敵対的生成ネットワークを用いた集団型異常検知

https://dbsj.org/wp-content/uploads/2020/02/DBSJ_18_2_maru.pdf

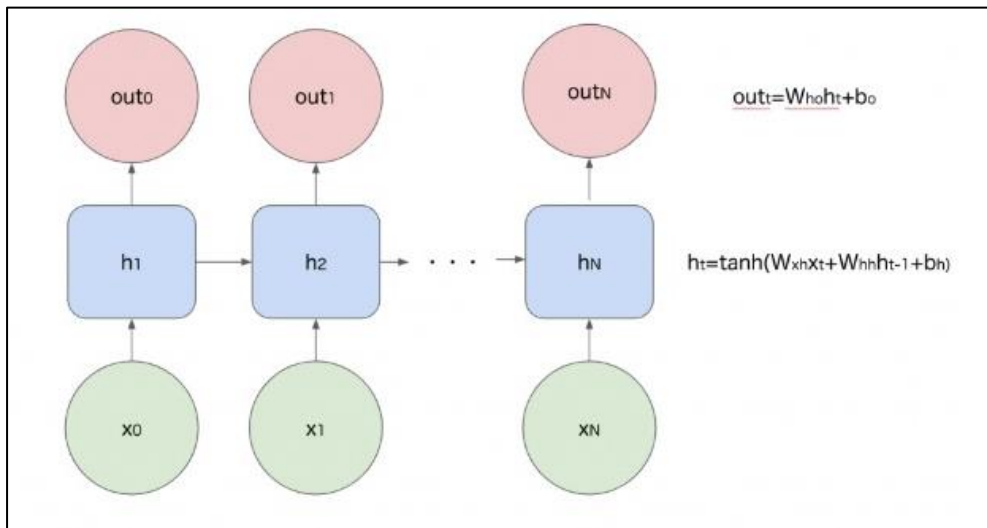
2. 今回の深層学習について

RNN がベースですが、LSTM という手法を使用します。さらに、異常検出手法である EncDec-AD を用いて、異常度のスコアを算出します。順番に、概要を説明します。

2.1 RNN (recurrent neural network) について

概要図は以下で、入力 x_t に対して、隠れ層と呼ばれる h_x を設定します。

出力 out_t は、 h_x を介して計算されるものです。さらに、 h_x は、 h_{x-1} からの影響を受けます。この場合、層が増すに従って、勾配が小さくなり、勾配消失問題が起きます。

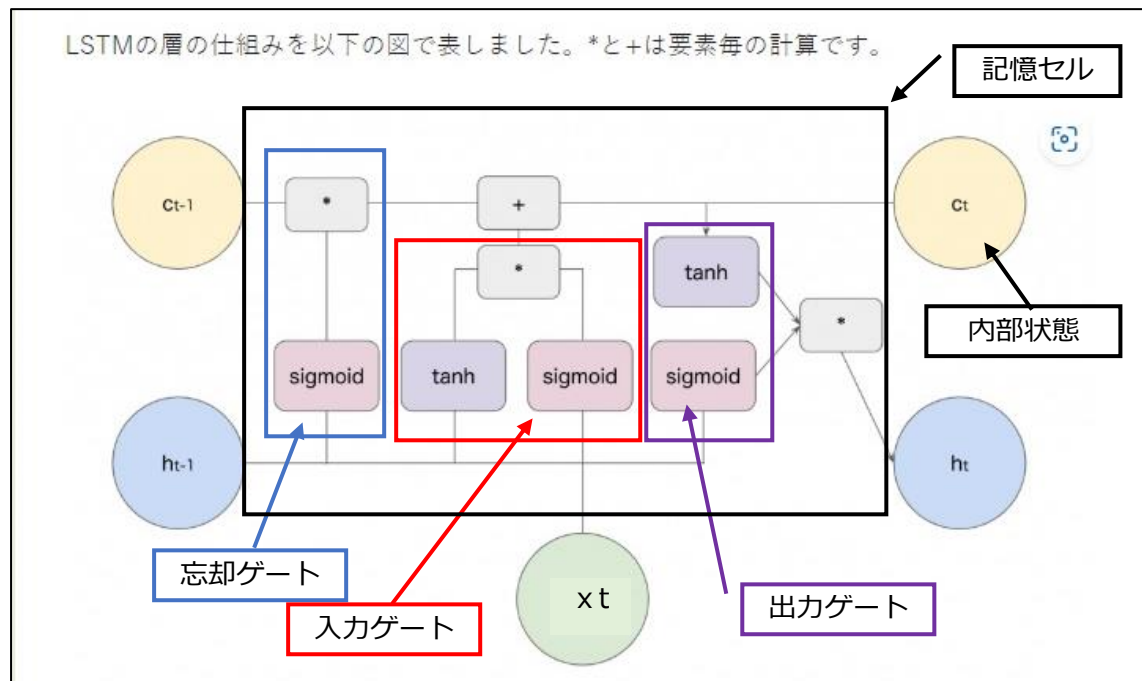


パラメーターを求める過程で、出力結果の誤差の総和を求めて、最小にする手法をとります。

具体的には、総和の勾配を最小にする点を探します。ただ、層が増えると、勾配が小さくなり、最小点が見つからなくなるというのが勾配消失問題です。

その対応策が、LSTM です。

2.2 LSTM(Long-Short Term Memory)について



sigmoid と tanh は活性化関数で、sigmoid は 0 から 1、tanh は -1 から 1 の値となります。

c は 内部状態を保持するメモリーに相当します。 $t-1$ から t へと 継承されていきます。

c_t は短期情報を長く保持するための状態変数 とされ、短期記憶 h_t を重み付けして記憶します。

c_t , h_t は、毎ステップ受渡されます。

忘却ゲートは、 c_t に対して、現在の入力ベクトル、短期記憶 h_t として、各ステップでどのくらい短期記憶を忘れ、どのくらい新たに短期記憶を保持すべきかを制御します。

入力ゲート、出力ゲートは、入力と出力を調整します。

上記を数式で表すと、以下となり、各パラメータの最適値を求めます。

$$\text{忘却ゲート } f_t = \sigma(W_f[x_t, h_{t-1}] + b_f)$$

$$\text{入力ゲート } i_t = \sigma(W_i[x_t, h_{t-1}] + b_i)$$

$$c_i = \tanh(W_c[x_t, h_{t-1}])$$

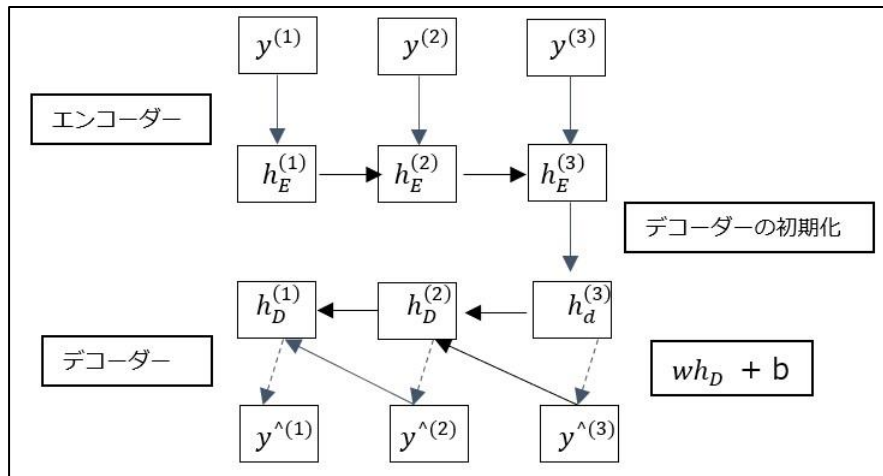
$$c_t = f_t * c_{t-1} + i_t * c_i$$

$$\text{出力ゲート } o_t = \sigma(W_o[x_t, h_{t-1}] + b_o)$$

$$h_t = o_t * \tanh(c_t)$$

Tensflow.Keras では、LSTM 関数が用意されているので、ブラックボックスとして処理できます。

2.3 EncDec-AD (Encoder/Decoder for Anomaly Detection) について



LSTM をエンコーダーとデコーダーとして用います。エンコーダーは特徴抽出(パターン抽出)、デコーダーは特徴から、推定値を求めます。

ウィンドウサイズ L の 部分系列データを $Y = \{y^{(1)}, \dots, y^{(L)}\}$ とします。 y は m 次元とします。

以下が 手順です。 学習データでまず行います。

1. Y を入力しエンコーダーで c 次元ベクトルである $h_E^{(L)}$ を計算します。
 $h_E^{(i)}$ ($i=1, \dots, L$) は、 $y^{(i)}$ を LSTM に入力し、 $h_E^{(i-1)}$ を更新することで得られます。
2. デコーダーの初期化として、 $h_D^{(L)} = h_E^{(L)}$ とします。
3. $h_D^{(i)}$ ($i=L, \dots, 1$)を用いて観測値を推定する。 推定値を $y^{(i)}$ とすると、以下となります。

$$y^{(i)} = wh_D^{(i)} + b$$

W は $m \times c$ 次元の全結合層の重み行列で、 b は m 次元のバイアスベクトル

4. y^i をデコーダーの LSTM に入力して $h_D^{(i-1)}$ を計算します。
5. 手順 3 および 4 を繰り返し、 $y^{(1)}$ まで 計算します。
6. $y^{(1)}$ 、 $y^{(1)}$ 間の損失を計算し、 損失が最小になるようパラメタを更新します。

$$\text{損失} = \sum_Y \sum_{i=1}^L \|y^{(i)} - y^{(i)}\|^2 \text{ で計算します。}$$

異常スコアの算出は以下となります。

1. 検証データにフィットする正規分布の μ と Σ を求めます。(最尤推定で)
2. 残差 $e^{(i)} = | y^{(i)} - y^{(i)*} |$ を計算します。
3. $y^{(i)}$ 入力時の異常スコアは以下で計算します。

$$a^{(i)} = (e^{(i)} - \mu)' \Sigma^{-1} (e^{(i)} - \mu)$$

閾値 の算出について

P 適合率 (Precision) と R 再現率 (Recall) は、トレードオフですので、

P と R の調和平均を取り、その値を F 値 (F-score) とし、最大になる閾値を調整します。

$$F_{\beta} = (1 + \beta^2) \times P \times R / \beta^2 P + R$$

ここで β は ハイパーパラメータ です。 ($\beta < 1$)

また、この値が高い上位 20%の観測値を異常と判定することもあるようです。

3 計算結果

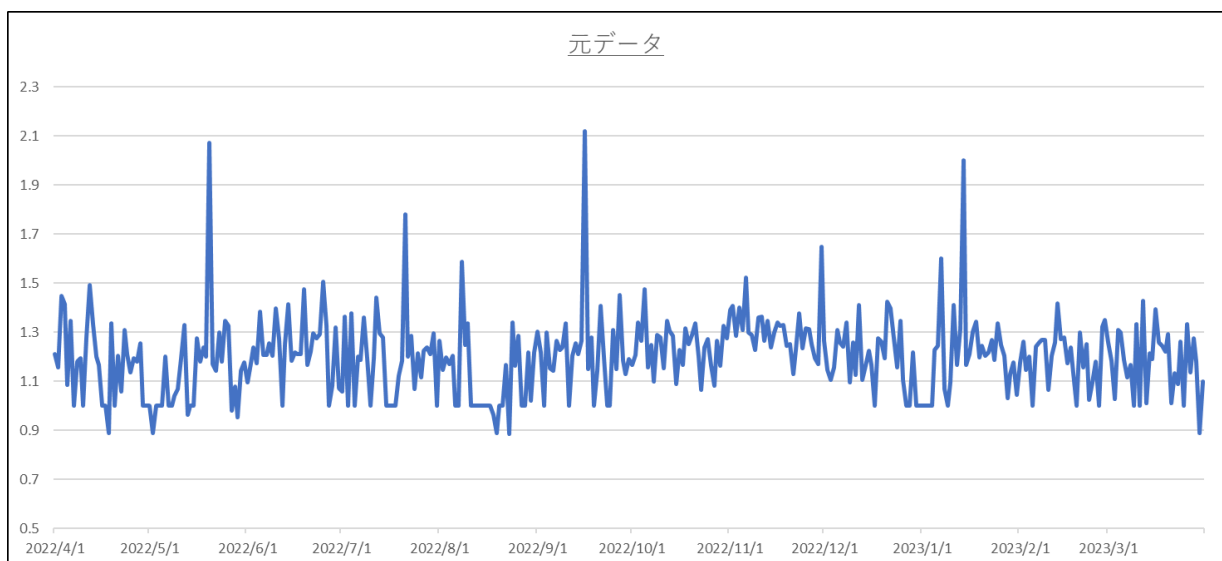
3.1 どのようなデータを用いたか、

放電加工機は、加工中に障害が発生すると自己診断して3段階のレベルに分け、レベルに応じたメッセージを表示します。内容は以下です。

- ・エラーメッセージ：続行不可能な障害が発生したときに表示し、動作を中断する。
- ・ハルトメッセージ：再開可能な障害が発生したときに表示し、一時停止する。
- ・コメントメッセージ：続行可能な障害で発生し、注意を促す。

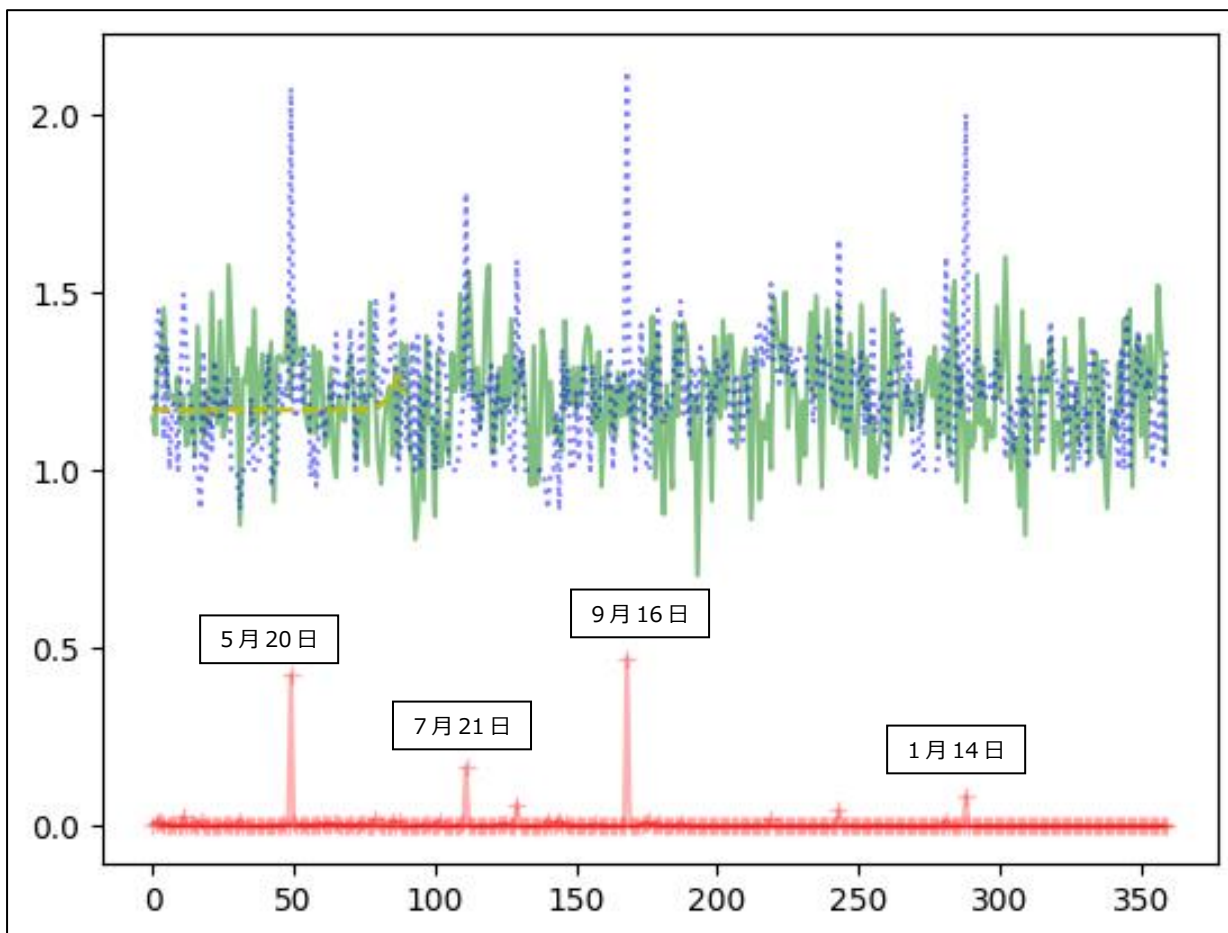
レベルに対応してデータ処理できるように、3段階のレベルについて障害のレベルが大きいと、点数は大きくなるように点数を設けました。

これらは、メッセージが発生した時の時間毎のデータで、点過程データと呼ばれるもので、時系列データではありません。故に、一日分の平均値を取って時系列データとして取り出すようしました。本データを以下に示します。2022年4月から2023年3月までのデータです。



3.2 異常検知のグラフ

(緑線：訓練データ 青破線：テストデータ 赤線：異常スコア)



5月20日 “テンション装置が異常です” メッセージが多発され、電源が落ちました。

放電加工機の再起動で復帰したようです。

7月21日 “テンション0調が異常です” のメッセージが多発しました。

9月16日 “(HCUT) AWT 動作エラーです” のメッセージが多発されてます。

この日は、メーカーの修理対応の日で、検証から出たメッセージです。

ここの異常度は、意図があるので、検証の対象にはなりません。

1月14日 “一時停止です” のメッセージがあり、自動結線の不調で、プログラムが停止しました。

4 まとめ

異常検出手法 EncDed-AD を実装して、プロットの結果を見ると、放電加工機の状態と 高めピーク値の発生時の間で相関があるようで、異常検知に有効と考えます。