ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Region of interest detection using MLP

Tommi Kärkkäinen[1] and Alexandr Maslov[1] and Pekka Wartiainen[1] *

Department of Mathematical Information Technology
P.O. Box 35, 40014 University of Jyväskylä - Finland

Abstract. A novel technique to detect regions of interest in a time series as deviation from the characteristic behavior is proposed. The deterministic form of a signal is obtained using a reliably trained MLP neural network with detailed complexity management and cross-validation based generalization assurance. The proposed technique is demonstrated with simulated and real data.

1 Introduction

Change point detection from a time series is defined as determination of those time stamps where statistical properties change significantly accordingly to some predefined criterion. Typical algorithms for this purpose rely on detecting changes in first or second order statistics, like mean, median, or standard deviation, or parameters of a statistical model of data distribution [1, 2, 3]. The challenge for real application, e.g., with industrial measurements [4], is that one needs to define and estimate the probability distribution which establishes the basis for change detection [5] (see also [6] for an overview). In this context, a region of interest (ROI) is considered as a subsequence containing one or more change points.

Traditionally, neural networks have been utilized with industrial time series data mainly for classification tasks [7]. MultiLayered Perceptron (MLP) neural networks are known to be universal nonlinear regression approximators [8]. However, for real applications this is just the beginning, as summarized by Hornik, Stinchcombe, and White [9]: "We have thus established that such 'mapping' networks are universal approximators. This implies that any lack of success in applications must arise from inadequate learning, insufficient numbers of hidden units or the lack of a deterministic relationship between input and target." Therefore, a reliable network training needs to address two other principal characteristics of a data based model in addition to its accuracy: *complexity* and *generalization to unseen data*.

Here we first describe an MLP training algorithm which takes into account these targets by a detailed management of network's *structural* (size of hidden layer) and *functional* (size of weights) complexity, targeting at highly reliable generalization using the well-known cross-validation technique [10]. This training framework is then applied to the given time series to train MLP capturing its deterministic behavior. To this end, those subintervals in time where there are significant deviations greater than predetermined threshold from the model's

*The authors gratefully acknowledge the support from Jenny and Antti Wihuri Foundation (TK) and from the OSER project (AM and PW).

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

predictions, are suggested as *ROIs*. We emphasize that the proposed technique is *unsupervised*, i.e., it does not utilize any labeling of ROIs, even if they are known in advance.

The contents of the rest of the paper are as follows: we describe the proposed method in Section 2, and report and conclude the computational experiments in Section 3.

2 The method

2.1 MLP Training

Action of MLP in a layer wise form can be given by (e.g., [11])

$$\mathbf{o}^0 = \mathbf{x}, \quad \mathbf{o}^l = \mathcal{F}^l(\mathbf{W}^l \tilde{\mathbf{o}}^{(l-1)}) \ \text{ for } l = 1, \ldots, L. \tag{1}$$

By $\tilde{}$ we indicate the vector enlargement for the bias and $\mathcal{F}^l(\cdot)$ denotes the activation function. This places biases in a layer as first column of the layer's weight matrix which then have the factorization $\mathbf{W}^l = \begin{bmatrix} \mathbf{W}_0^l & \mathbf{W}_1^l \end{bmatrix}$.

Using the given learning data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^{n_0}$ and $\mathbf{y}_i \in \mathbb{R}^{n_L}$, the unknown weight matrices $\{\mathbf{W}^l\}_{l=1}^L$ in (1) are determined as a solution of an optimization problem

$$\min_{\{\mathbf{W}^l\}_{l=1}^L} \mathcal{J}(\{\mathbf{W}^l\}). \tag{2}$$

Here we restrict ourselves to MLP with one hidden layer and our cost functional reads as follows:

$$\mathcal{J}(\{\mathbf{W}^1, \mathbf{W}^2\}) = \frac{1}{2N} \sum_{i=1}^{N} \left\| \mathbf{W}^2 \tilde{\mathcal{F}}^1(\mathbf{W}^1 \tilde{\mathbf{x}}_i) - \mathbf{y}_i \right\|^2 + \frac{\beta}{2n_1} \sum_{(i,j)} \left(|\mathbf{W}_{i,j}^1|^2 + |(\mathbf{W}_1^2)_{i,j}|^2 \right) \tag{3}$$

for $\beta \geq 0$. The special form of regularization omitting the bias-column $\mathbf{W}_0^2$ is due to Corollary 1 in [12]: *Every locally optimal solution to* (2) *provides an unbiased regression estimate having zero mean error.*

The universal approximation property guarantees accuracy of an MLP network, but in practical applications we also need to address *simplicity* and *generalization*. Simplicity is further divided into *structural simplicity*, which means favoring small size of the hidden layer, and *functional simplicity*, which refers to favoring small weights improving the network's fault tolerance [13]. Hence, in our actual training method we use a grid search for both size of the hidden layer n_1 and size of the regularization coefficient β. Moreover, 10-fold cross-validation is used as a technique to assure proper generalization of the obtained MLP network. To this end, the usual gradient based optimization methods for minimizing (3) act locally and, therefore, the solution depends on the initialization. In order to explore the search landscape better towards global optimization, we repeat the random started optimization solver three times and select, as the solution, the one with minimal training error. In the final training of MLP with fixed n_1 and β, we test five iterations for slightly more thorough globalization.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Algorithm 1 Reliable determination of MLP neural network.

Input: Training data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$.
Output: MLP neural network.
Define $\vec{\beta}$, $n1max$, and $nfolds$
Create $nfolds$ using random sampling
for $n_1 \leftarrow 1$ **to** $n1max$ **do**
  for $regs \leftarrow 1$ **to** $|\beta|$ **do**
    for $k \leftarrow 1$ **to** $nfolds$ **do**
      for $i \leftarrow 1$ **to** 3 **do**
        Initialize $(\mathbf{W}^1, \mathbf{W}^2)$ from $\mathcal{U}([-1, 1])$
        Minimize (3) with current n_1 and $\vec{\beta}(regs)$, and the CV_{Tr}
        Store Network for smallest $P_{err,Tr}$
      Compute $P_{err,Te}$ for the stored Network
    Store $n_1^* = n_1$ and $\beta^* = \beta$ for the smallest mean $P_{err,Te}$
for $i \leftarrow 1$ **to** 5 **do**
  Initialize $(\mathbf{W}^1, \mathbf{W}^2)$ from $\mathcal{U}([-1, 1])$
  Minimize (3) using n_1^*, β^* and the whole training data
  Select the network with smallest P_{err}

Minimization of (3) is based on MATLAB's unconstrained minimization routine *fminunc*[1] with self realized MLP cost function and gradient calculations along the lines of [12] (these are done in full matrix form and then reshaped to and from one long weight vector for the optimizer). The vector of regularization parameters is defined as $\vec{\beta} = 10^{-i}, i = 1, \ldots, 6$. The prediction error ($P_{err,[Tr|Te]}$) is computed as the mean Euclidian error. In MLP, the sigmoidal activation function $s(x) = \frac{1}{1+\exp(-x)}$ is used. Moreover, all input and output variables are preprocessed into the range $[0, 1]$ of $s(x)$ to balance their scaling with each other and with the range of the overall transformation [12].

2.2 Application of MLP for ROI detection

Assume that a time series $\{s(t_i)\}_{i=1}^{T}$ is given. The learning data for MLP is created in the usual way: first a window length $L \in \mathbb{N},\ L > 1$, is fixed and then we associate $y_i = s(t_i), i = L, \ldots, T$, and $\mathbf{x}_i = \{s(t_{i-j}),\ j = L-1, \ldots, 1\}$.

Then Algorithm 1 is applied to train a reliably generalizing network capturing the deterministic behavior of the time series. With this model, the absolute prediction error time series

$$e_i = |\mathcal{N}(\{\mathbf{W}^l\})(x_i) - y_i|, \quad i = L, \ldots T,$$

is created. To this end, a threshold $\tau \in \mathbb{R}$ is fixed and those indices, for which $e_i > \tau$, are proposed as members of ROIs.

[1] *http://www.mathworks.se/help/optim/ug/fminunc.html*

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

3 Experimental results and conclusions

We illustrate the proposed algorithm using two examples, a simulated and a real data set one. The threshold τ is set to 0.05 and $L = 8$ is used as the data window size. We use separate learning data and validation data to assess the method's performance to detect ROIs.

Example 1 *We created a simulated case with sinusoidal wave form and added normally distributed degradations of different strength to four subregions. Similar form with three noncharacteristics subregions is used as validation data. (see Figure 1)*

Example 2 *We use the Dodgers data set from UCI repository*[2]*. The data describes a five minute sampled traffic sensor reading storing the amount of cars passing a ramp on a freeway in Los Angeles. The learning problem is to determine the times of football games which are provided in another file. In the whole data there is almost six months of measurements (10-Apr-2005 00:00 – 01-Oct-2005 23:55), but occasionally the sensor is off. From the measurements, we used the indices {380–2466} (11-Apr-2005 07:35 – 18-Apr-2005 13:25, first five matches) as training data filtering out periods where the sensor is off. As validation data, the indices {3284-12529} (21-Apr-2005 09:35 – 23-May-2005 12:00, next 18 matches) are used.*

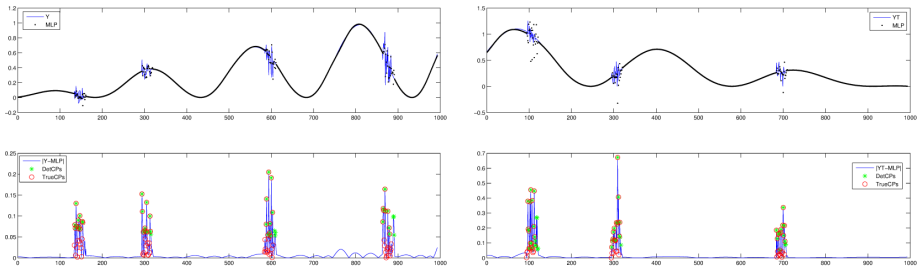

Fig. 1: Detected change regions for sinusoidal training data (left) and test data (right).

In the result figures (Figures 1,2,3), real change points in ROIs are depicted with red circles and, correspondingly, ROI indices proposed by the thresholded MLP error are given by green stars. Because Example 2 is related to traffic near the football stadium, a reasonable assumption is to assume uncharacteristic traffic patterns also before and after the actual match times. Therefore, we accept as correct indication of ROI one hour before and after the game.

Because deviation from a normal behavior of a signal can correspond to noise or actual change, we assume that the time series is not dominated by noise.

[2] *http://archive.ics.uci.edu/ml/datasets/Dodgers+Loop+Sensor*: "These loop sensor measurements were obtained from the Freeway Performance Measurement System (PeMS)"

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Therefore, in the Dodgers example we first apply mean filtering with window size 11 for denoising. The filter size of 11 is selected empirically in order to achieve smallest window size for removing noise but sustaining real behaviour of the data.

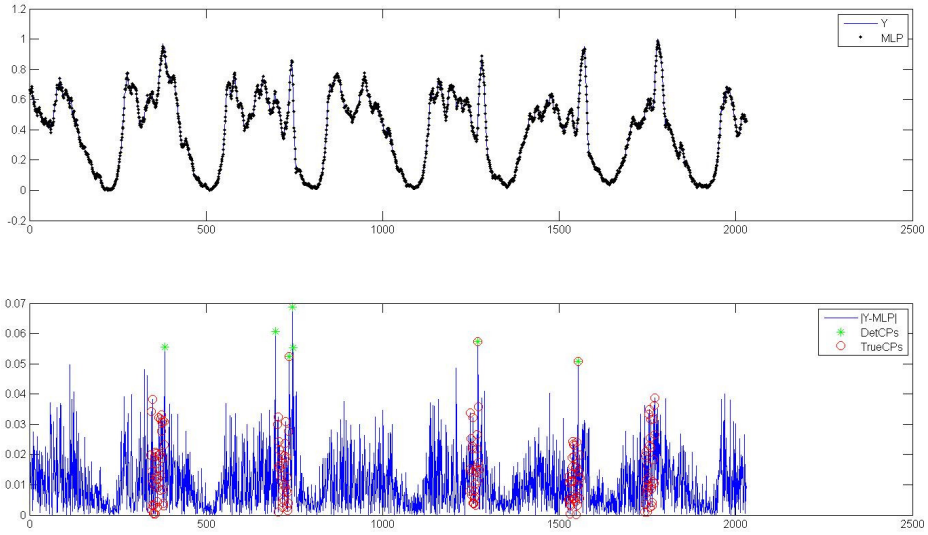

Fig. 2: MLP training compared to detected ROIs for Dodgers training data.

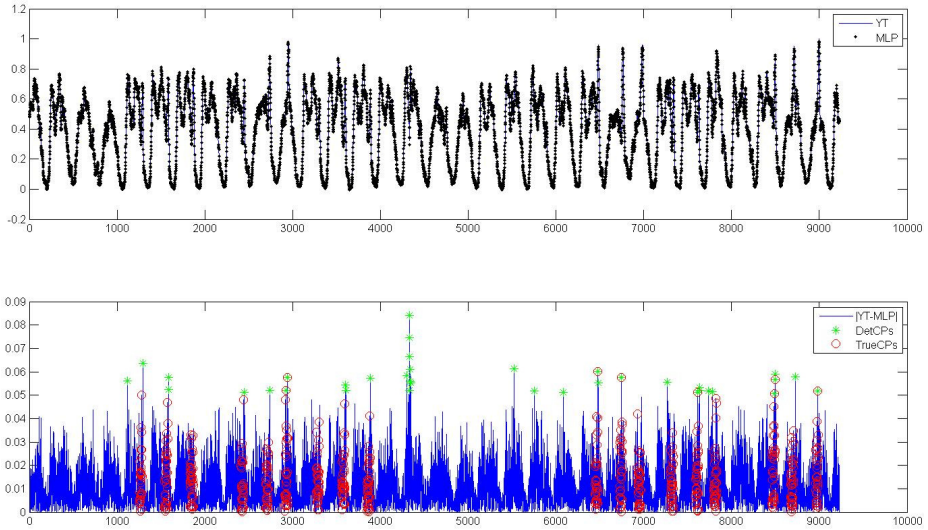

Fig. 3: MLP testing compared to detected ROIs for Dodgers test data.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

From the figures we conclude that changes from normal deterministic behaviour are detected very well with the simulated case, both with training and validation data (Figure 1). Also, preliminary results are promising with real data. In the training data, 4 of 5 games were detected (Figure 2). With this trained network, 14 out of 18 ROIs were successfully located from the validation data (Figure 3) with 6 false-positive alerts. However, this was the best result obtained after several runs of the overall algorithm, which is not fully deterministic due to random creation of folds in cross-validation.

Our numerical experiments confirm the potential of the proposed approach. When the characteristic behavior of a time series is smooth and deviation clearly visible, as in Example 1, the results are as expected. Even if no such separation exists, we were able to identify potential and in many cases correct ROIs in Example 2. As can be seen, however, in such cases it might be difficult to say whether a noisy behavior (even after denoising) or actual change regions are captured. Therefore, reliable denoising is a prerequisite for good performance of the approach. The method could be improved, e.g., by feature extraction to replace the raw time series values as MLP input.

References

[1] Igor V. Nikiforov and Michèle Basseville. Detection of Abrupt Changes - Theory and Application, 1998.

[2] Nicolas Cheifetz, Allou Samé, Patrice Aknin, and Emmanuel de Verdalle. A cusum approach for online change-point detection on curve sequences. *Computational Intelligence and Machine Learning, Bruges (Belgium)*, pages 25–27, 2012.

[3] Marc Lavielle. Using penalized contrasts for the change-point problem. *Signal Processing*, 85(8):1501 – 1510, 2005.

[4] Luigi Fortuna, Salvatore Graziani, Alessandro Rizzo, and Maria G. Xibilia. *Soft Sensors for Monitoring and Control of Industrial Processes*. Spronger, 2007.

[5] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 180–191. VLDB Endowment, 2004.

[6] João Gama. *Knowledge Discovery from Data Streams*. Chapman and Hall / CRC Data Mining and Knowledge Discovery Series. CRC Press, 2010.

[7] Silvio Simani and Ronald J. Patton. Neural networks for fault diagnosis of industrial plants at different working points. In Michel Verleysen, editor, *ESANN*, pages 495–500, 2002.

[8] Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, pages 143–195, 1999.

[9] Kurt Hornik, Maxwell Stinchcombe, and Halber White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[10] Larson S. The shrinkage of the coefficient of multiple correlation. *Journal of Educational Psychology*, 22:45–55, 1931.

[11] M. T. Hagan and M. B. Menhaj. Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Networks*, 5:989–993, 1994.

[12] Tommi Kärkkäinen. MLP in layer-wise form with applications in weight decay. *Neural Computation*, 14:1451–1480, 2002.

[13] Salvatore Cavalieri and Orazio Mirabella. A novel learning algorithm which improves the partial fault tolerance of multilayer neural networks. *Neural Networks*, 12:91–106, 1999.