IDENTIFICATION OF SENSOR FAULTS ON COMBINE HARVESTERS USING INTELLIGENT METHODS

Geert Craessaerts, Tom Coen and Josse De Baerdemaeker

Laboratory for Agro-Machinery and Processing, KULeuven, Kasteelpark Arenberg 30, B-3001 Leuven, Belgium; E-mail: Geert.Craessaerts@agr.kuleuven.ac.be

Abstract: Process monitoring and fault diagnosis is of considerable interest from an industrial perspective. In this paper, the general applicability of intelligent methods, like self-organizing maps (SOM) and multilayer feedforward networks with backpropagation , for the identification of sensor failure on combine harvesters will be illustrated. Both neural network types showed comparable results in order to classify normal and faulty sensor conditions. *Copyright © 2005 IFAC*

Keywords: fault identification, self organizing system, feedforward networks, backpropagation algorithms, agriculture.

1. INTRODUCTION

Constructors of agricultural machinery always strive for the development of more efficient and performant products. Accordingly, big investments are dedicated to the development of new sensors in order to gain some process knowledge. Considering the automation and control of agricultural machinery, condition monitoring of machines is gaining importance in industry because of the need to increase reliability and to decrease possible loss of production due to machine breakdown. Hence, there is considerable interest in this field now from an industrial perspective. There is an abundance of literature on process fault diagnosis ranging from analytical methods to artificial intelligence and statistical approaches (Venkatasubramanian, et al., 2003).

Neural networks have emerged as a powerful tool for pattern recognition. Like other pattern recognition techniques, neural networks act on data by detecting some kind of underlying organization. The networks can recognize spatial, temporal, or other relationships and can perform such tasks as classification, prediction and function estimation.

In this study, the general applicability of supervised self-organizing maps (SOM) and back-propagation neural networks for the identification of sensor failure will be illustrated by an off-line evaluation of both intelligent methods on a New Holland CX combine harvester. By comparing the signals of a machine running in normal and faulty conditions, detection of sensor failure becomes possible.

2. THE COMBINE HARVESTING PROCESS

Combine harvesters are large, complex machines sent out to all corners of the world to harvest different types of crops under all possible environmental conditions. The working process of a combine harvester can be divided into four different sub-processes: 1, cutting of crop systems and collection of harvested material; 2, the separation of grain kernels from larger crop parts as straw; 3, the cleaning process, separating grain kernel from other small particles as chaff and short straw; and 4, the temporary storage of clean crop material in the grain bin.

The combine harvesting process is a highly uncertain process asking for sensors that extract immediate information from the process. Different measurement devices are installed to extract information from the combine process with an eye to the automation of the combine process. The sensors used in this study were installed on a New Holland CX test combine (Fig. 1) and were read out during field experiments with the *CANbus Control Design Interface* program, developed for the on-the-go design and evaluation of combine automation systems.

Fig. 1. New Holland CX840 combine harvester harvesting in wheat.

The sensors used in this study are:

- *Machine speed:* The rotation speed of the driving wheels is commonly used to estimate ground speed.

  Feedrate sensor: Measures the driving torque of the header as a measure of the total crop flow into the harvester
- *Grain Pan load sensor:* a sensor according to the principle of De Baerdemaeker (1989), to measure the crop load on the the grain pan.
- *Walker Loss*: An impact sensor, installed at the end of the walker section to estimate the amount of residual grain kernels between the straw particles.

3. SELF-ORGANIZING MAPS (SOM)

The SOM is a neural network methodology developed by Kohonen that forms a two-dimensional presentation from multi-dimensional data (Kohonen, 1995). During this transformation, the topology of the data is kept in the presentation such that data vectors, which closely resemble one another, are located next to each other on the map. An important characteristic of the SOM is generalization of the information, which enables the classification of data vectors not used in the training of the SOM. The SOM can thus serve as a clustering tool of high-dimensional data which were not included in the training data set.

Each neuron i of the SOM is represented by an n-dimensional weight, or model vector, $\mathbf{m}_i=[m_{i1},\ldots,m_{in}]^T$ (n is the dimension of the input vectors). The neurons are connected to adjacent neurons by a neighbourhood relation, which dictates the topology, or structure of the map.

3.1 Training of the SOM

During iterative training procedure, the SOM forms an elastic net that folds onto 'cloud' formed by input data. The net tends to approximate the probability density of the data (Kohonen, 1995). The model vectors tend to drift there were the data are dense, while there are only a few model vectors where data are sparse. For the training of the SOM there exist both batch versions and sequential or on-line adaptive versions.

<u>Sequential training algorithm</u>

The SOM is trained iteratively. In each training step, one sample vector x from the input data set is chosen randomly and the distances between it and all the weight vectors of the SOM are calculated using some distance measure. The neuron whose weight vector is closest to the input vector is called the Best-Matching Unit (BMU) denoted here by c:

$$\| x - m_c \| \;=\; \min_i \; \{\| x - m_i \|\} \qquad (1)$$

where $\|\cdot\|$ is the distance measure, typically Euclidean distance. After finding the BMU, the weight vectors of the SOM are updated so that the BMU is moved closer to the input vector in the input space. The topological neighbours of the BMU are treated similarly. The SOM update rule for the weight vector of unit i is:

$$m_i(t+1) = m_i(t) + a(t)h_{ci}(t)[x(t) - m_i(t)] \qquad (2)$$

where t denotes time. The x(t) is an input data vector randomly drawn from the input data set at time t, $h_{ci(t)}$ the neighbourhood kernel around the winner unit c and $\alpha(t)$ the learning rate at time t. The training is usually performed in two phases. In the first phase, relatively large initial learning rate α_0 and neighbourhood radius σ_0 are used. In the second phase both learning rate and neighbourhood radius are small right from the beginning. This procedure corresponds to first tuning the SOM approximately to the same space as the input data and then fine-tuning the map.

<u>Batch training algorithm</u>

The batch algorithm is also iterative, but instead of using a single data vector at a time, the whole data set is presented to the map before any adjustments are made. In each training step, the data set is partitioned according to the Voronoi regions of the map weight vectors, ie. each data vector belongs to the data set of the map unit to which it is closest. After this, the new weight vectors are calculated as:

$$m_i(t+1) = \frac{\sum_{j=1}^{n} h_{ic}(t)x_j}{\sum_{j=1}^{n} h_{ic}(t)} \qquad (3)$$

where $c = \arg\min_k \{\|x_j - m_k\|\}$ is the index of the BMU of data sample x_j. The new weight vector is a weighted average of the data samples, where the weight of each data sample is the neighbourhood function value $h_{ic}(t)$ at its BMU c.

4. MULTILAYER FEEDFORWARD NETWORKS WITH BACKPROPAGATION

Feed-forward neural networks (Rumelhart, et al., 1986) provide a general framework for representing non-linear functional mappings between a set of input and output variables. Such a network consists of one or more hidden layers and an output layer. A multilayer perceptron with one hidden layer is shown in Fig. 2.

The key to the back propagation algorithm is its ability to change the values of its weights in response to errors. For it to be possible to ca lculate the errors, the training data must contain a series of input patterns labeled with their target output patterns. During back propagation training, the network passes the derivatives of the output errors back to the hidden layer, using the original weighted connections. This backward propagation of errors gives the algorithm its name. Each hidden node then calculates the weighted sum of the back-propagated errors to find its indirect contribution to the known output errors. After each output and hidden node finds its error value, the node adjusts its weights to reduce its error.

The Levenberg-Marquardt (LM) algorithm is the most widely used optimization algorithm. The problem for which the LM algorithm provides a solution is called *Nonlinear least squares minimization (Hagan, et al., 1996).* This implies that the function to be minimized is of the following special form:

$$f(x;w) = \frac{1}{2}\sum_{j=1}^{m} r_j^{\,2}(x;w) \qquad (4)$$

where r_j is a function from R^n to R. Each r_j is referred as residual or error between the actual results and target results. The w represents the weights or parameters that has to be optimized.

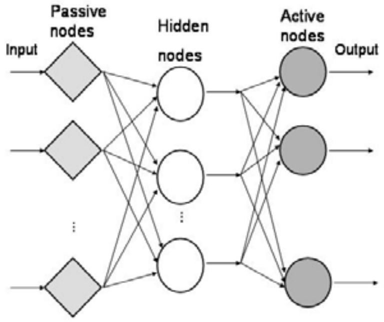

Fig. 2. Multilayer perceptron with one hidden layer

To find the minima of the function f, Levenberg proposed an algorithm whose update rule for the weights is a blend of the gradient descent and Gauss-Newton iteration algorithms.

$$w_{i+1} = w_i - (H + \lambda I)^{-1}\nabla f \qquad (5)$$

where H is the Hessian matrix and ∇f are the derivatives of *f.*
Marquardt replaced the identity matrix in equation 5 with the diagonal of the Hessian resulting in the Levenberg-Marquardt update rule.

$$w_{i+1} = w_i - (H + \lambda diag[H])^{-1}\nabla f \qquad (6)$$

5. RESULTS AND DISCUSSION

The neural network approach for sensor fault identification was applied on a dataset gathered with a conventional CX combine harvester during wheat harvest. A dataset of 44163 samples was registered at a sample rate of 5 Hz. In order to achieve a good end result, the following pre-processing steps were applied to the dataset:

- Analog filtering of the data by a second order low-pass filter with a cut-off frequency of 0.15 Hz.
- Normalization of the data
- Labelling of the data: Data were subdivided into three different regimes; 1, normal operation; 2, failure of a grain pan load sensor; 3, failure of the walker loss sensor

The data set was at random splitted in a training (2/3) and test set (1/3). This selection procedure of training and test set along with the training and testing of the different neural network model structures was performed a number of times (20 cycles) in order to cope with the intrinsic variability of neural networks (selection of data, initialization of weights,…). After pre-processing of the data, an evaluation was made of both intelligent methods (self-organizing maps and backpropagation neural networks) for the identification of sensor failure.

5.1 *Sensor fault detection by supervised Self-organizing maps*

Different SOM configurations and training algorithms were evaluated for detection of sensor failure. In this section, an overview will be given of the performances of the different map configurations and training algorithms.

In a first step of the training procedure, the SOM was initialized linearly. The initialization was made by first calculating the eigenvalues and eigenvectors of the given data. The map weight vectors were initialized along the two greatest eigenvectors of the covariance matrix of the training data. After initialization, the map was trained on a supervised way by both sequential and batch algorithms. The training length, 10 epochs , was held constant during the tests. A hexagonal lattice and gaussian neighbourhood function was used during all training procedures.

The result of a sequential training procedure of a 5*5 self-organizing map is shown in Fig.3.

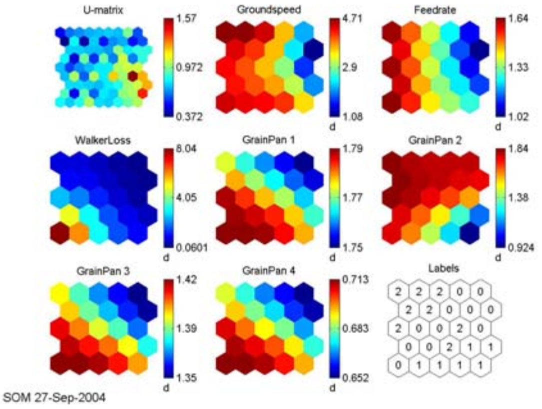

Fig. 3. Visualization of a SOM trained by a sequential training algorithm

The U-matrix is shown along with the seven component planes and label grid. By making use of the U-matrix, the distances between each map unit to each of its immediate neighbours is calculated. Distance matrix methods show the borders between the different clusters. Each component plane consists of the values of a single vector component in all map units. The neurons of the SOM were labelled based on a voting procedure and labels of the training data set; '0' label corresponds with normal operating regime, '1' label corresponds with a failure of a grain pan load sensor, '2' label corresponds with a failure of the walker loss sensor.

Different map configurations and training procedures were tested in the field of an accurate classification of new test data. Test data were classified into one of the three different regimes by making use of an auto-labelling algorithm. This algorithm assigns a label to the new test data by locating the new data vector on the trained label grid. The classification of test data was classified as correct if the label, assigned by the auto-labelling algorithm, corresponds with the real label, assigned during the pre-processing phase.
The results are illustrated in Fig. 4.

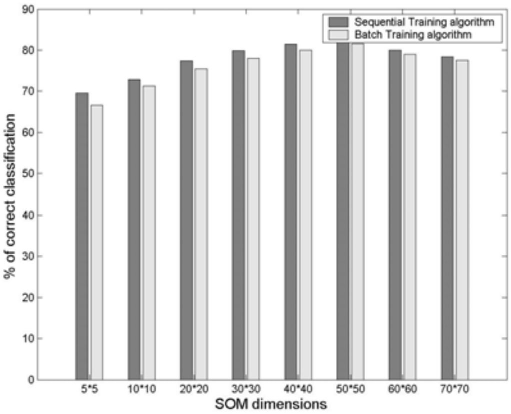

Fig. 4. Classification performance of different SOM configurations and training algorithms

Analysis of the results in Fig. 4 shows that:

- A sequential training algorithm achieves better classification results in comparison with batch algorithms.
- The classification performance increases with the map size until overtraining occurs. This occurred at a map dimension of 60*60 and implies less generalisation power.
- Best classification accuracy (82.37%) was achieved with a 50*50 SOM and a sequential training algorithm.

5.2 *Sensor fault detection by multilayer feedforward neural networks with backpropagation*

Different multilayer perceptron architectures were evaluated for detection of sensor failure. During all the tests, a neural network with one input layer, one hidden layer and one output layer was used. The input layer had seven nodes representing the seven input signals. The number of neurons in the hidden layer was varied from 2 to 14. The number of output nodes, two, was constant during the tests. The target values of the output nodes could have only binary levels representing normal and failed bearing. During the training stage, the target value of the two output nodes for the normal state conditions were set 0. For failed grain pan sensor conditions, the first output node was set 1 and the second output node was set 0. Similarly for failed walker loss sensor conditions, the first output node was set 0 and the second output node was set 1. The artificial neural network was trained and implemented using the MATLAB neural network toolbox using backpropagation with Levenberg-Marquardt algorithm.

The classification performance of different neural network architectures, considering new test data, is illustrated in Fig. 5 and a selection of the most important results is shown in Table 2. A maximum number of 14 hidden neurons was used during the test due to excessive training times with more complex architectures.

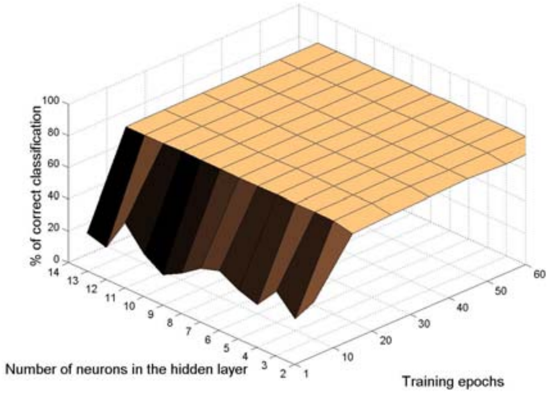

Fig. 5. Classification performance of different multilayer perceptron architectures.

Table 2: Percentage of correct classification of different process regimes by making use of multilayer perceptrons

Number of hidden layer neurons	10 training epochs	30 training epochs
2	77.62	79.29
4	80.34	80.85
6	80.67	81.01
8	80.49	81.08
10	80.78	81.14
12	80.74	80.98
14	80.65	80.98

Analysis of the results in Fig. 5 and Table 2 shows that:

- No significant increase of classification performance was reached by an increase of the number of neurons in the hidden layer.
- No significant increase of classification performance was reached by an increase of the training epochs.
- No overtraining occurred during the tests.
- A classification performance of approximately 80% was reached with a rather simple network (4 hidden neurons) and training procedure (10 training epochs).

6. CONCLUSION

An artificial neural network procedure was presented for sensor fault diagnosis in combine harvesters. Two intelligent methods, namely supervised self-organizing maps and multilayer feedforward neural networks with backpropagation were evaluated within the framework of the development of a supervisory fault detection system. Different configurations of both neural network types were several times trained by a random training set and their classification performance was tested. Both neural network types showed comparable classification results for new test data. However, best results were obtained with a supervised self-organizing map of 50*50 dimension. With this neural network configuration, it became possible to classify 82% of the new test data correctly. Self-organizing maps offer also interesting tools within the framework of monitoring and visualization of the different process regimes.

According to the theory, unsupervised neural network models can be constructed solely on the basis of measurement data of the process. However, if supervised neural networks are used, like in this study, information about the process state is needed. This demonstrates that, despite new modelling techniques, process knowledge still plays an important role in fault detection systems.

ACKNOWLEDGEMENTS

The author gratefully acknowledges ‘Instituut voor de aanmoediging van Innovatie door Wetenschap en Technologie in Vlaanderen’ (IWT) for the support. This study was made possible through cooperation of CNH Belgium N.V.

REFERENCES

De Baerdemaeker, J., A.J. Srivastava and P. Lindemans (1989). Measuring the material flowrate on the preparation table of a combine. *ASAE meeting paper, Louisiana.* **89-1591**, 7 pages.

Hagan, M.T., H.B. Demuth and M. Beale (1996). *Neural network design.* PWS Publishing Company, Boston.

Kohonen, T. (1995). *Self-Organizing Maps*. Springer-Verslag, Berlin.

Rumelhart, D.E., G.E. Hinton and R.J. Williams, (1986). Learning internal representations by error propagation. In: *Parallel Distributed Processing (Part 1)*, (Rumelhart, D.E., J.L. McClelland. (Eds.)), pp. 318–362. MIT Press, Cambridge, USA.

Venkatasubramanian, V., R. Rengaswamy, S.N. Kavuri and K. Yin (2003). A review of process fault detection and diagnosis Part III: Process history based methods. *Computers and chemical engineering,* **27**, 327-346.