IDIAP RESEARCH REPORT

IDIAP

Martigny - Valais - Suisse

Minimizing the Training Set for Image Classification applied to Quality Control

Miguel Moreira [a] Emile Fiesler [a]
Gianni Pante [b]

IDIAP–RR 96-10

January 1996

published in
Proceedings ISAI /IFIS, Cancun, Mexico, November 1996

Dalle Molle Institute for Perceptive Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[a] IDIAP, CP 592, CH-1920 Martigny, Switzerland
[b] GPIL, Rue du Grand Verger 6, CH-1920 Martigny, Switzerland

IDIAP Research Report 96-10

Minimizing the Training Set for Image Classification applied to Quality Control

Miguel Moreira Emile Fiesler Gianni Pante

January 1996

published in
Proceedings ISAI /IFIS, Cancun, Mexico, November 1996

Abstract. A method is presented for the automatic time detection of watches, where the hands are classified by a neural network. In order to reduce the overall cost of data collection, strict limits were imposed on the data collection time. This constraint severely limits the available amount of images, and poses the challenge to solve the hand recognition problem with a minimum amount of training and test data. Two neural network approaches are presented together with their performance results, which show an excellent final recognition rate.

1 Introduction

This document describes the procedures to be used in the Neural Network Module of a prototype of a system for watch quality control. The system is to be applied in the final part of the production line and uses Automatic Image Recognition techniques and Neural Networks for automatic time detection. The descriptions given here concern the Neural Network part. Details are also given about the interface established with the image technology used, as well as a general description of all the processing tasks followed by the prototype.

The details presented are based only on the watch models present in the catalogue of the particular watch maker for whom the project is to be developed. Although the solution described should apply to several other types of watches, it may not handle watches with special features not discussed here.

2 The case

The main task of the Quality Control System (QCS) is to automatically determine the exact time represented by an analog-type wrist watch. To accomplish this, a system involving a sequence of processing steps is used. The scheme of figure 1 illustrates the steps followed.

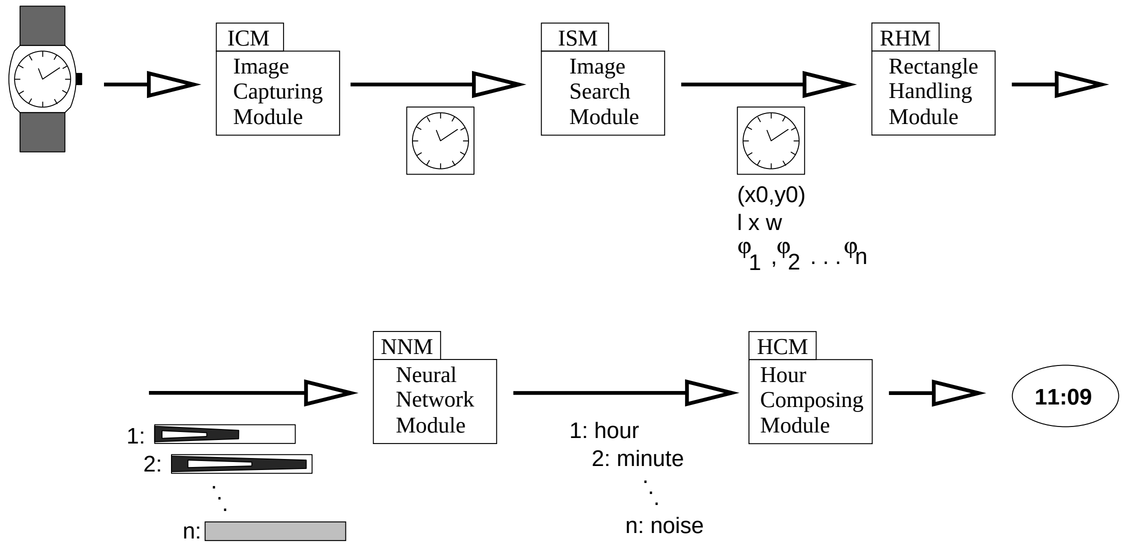

Figure 1: The sequence of processing steps followed by the QCS to recognize the time

First, the image of a watch is captured by the IMAGE CAPTURING MODULE (ICM), which includes a device to hold the watch, a CCD camera, and an associated controlling device. The captured image is passed on to an IMAGE SEARCH MODULE (ISM) that, using specific image handling techniques, scans it radially to find candidate positions of the HOURS hand and MINUTES hand, that is, the radial positions where each of the hands is most probably located. More than one candidate position are usually found for each hand. This module also does a search for the SECONDS hand position, which can be found accurately. The hands will henceforth be given the denomination of H-hand, M-hand, and S-hand, for the HOURS, MINUTES, and SECONDS hands, respectively. The image and the information associated with it, including the coordinates of the centre of the watch (x_0, y_0), the dimensions of the M-hand $(l \times w)$, and the set of N candidate positions angles $(\varphi_n)_{n\in[1..N]}$, are passed to the RECTANGLE HANDLING MODULE (RHM), that cuts rectangles of size $l \times w$, starting at the centre of the watch image and along the corresponding angle, as the scheme in figure 2 shows, and scales them to a fixed size. The scaling is done in order to normalize the input size for the Neural Network, located in the NEURAL NETWORK MODULE (NNM), that should decide, for each candidate, if the rectangle contains an HOURS hand, a MINUTES hand, or no hand at all (noise, which includes a possibly existing SECONDS hand or other alternative hands). The information containing the candidate angle values and the contents of the image at the corresponding positions are sent to the HOUR COMPOSING MODULE (HCM) that outputs the exact hour.

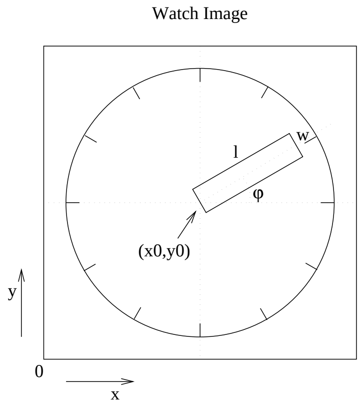

Figure 2: The cutting of a rectangle from an image

The image obtained by the ICM is in gray-scale TIFF format. This same format is also used by the ISM that does not introduce any changes in it. Only in the RHM is the image transformed into a matrix of 8-bit gray-scale values, before the rectangles are extracted and scaled. This 8-bit format is also used in the cut rectangles.

The description given so far concerns the processing sequence used to determine the time of a watch. Some of the referred modules need, however, to be trained in advance in order to correctly accomplish the desired tasks. This is the case for the IMAGE SEARCH and NEURAL NETWORK modules, that need to adapt their behaviour according to the specific watch type handled. The description of the NNM training procedure is given here. No details are given about the training of the ISM. A remark must be made concerning the fact that the training of both modules should be coordinated somehow, in order to avoid certain human-operator tasks common to both to be performed twice.

Due to practical reasons, the training is subject to two fundamental conditions: it should be fast and easy from the human operator point of view. This results in an important constraint: the operator will set the watch time only once before putting it definitively on the ICM, the process being totally automatic thereafter. The possibility of a second operator-based time setting during the training process should not be considered. This means that all the training images will be captured from a certain time range, that should not be large, given that the watch hands will move in real-time speed. The later constraint forces the need to minimize the total number of images used in the training. The actual capturing procedure is explained below while the choice of the appropriate time range is discussed later.

The training of the NNM may be divided into two parts. The first involves capturing the training images and composing the training sets, the second is the actual Neural Network (NN) training. For the first part, the operator selects a properly functioning watch for training. The watch is adjusted to a precise initial time ($T = 0$) and put in the ICM. The first task of the training consists in using the ISM features for the selection of the image centre coordinates and rectangle dimensions, followed by the search for the exact angular positions φ_H and φ_M of the H-hand and M-hand, respectively, at $T = 0$, given that the operator is conceded a certain angular tolerance in the initial hour setting and because the watch position on the ICM could be somewhat skewed with respect to the perfect orientation (with the 3h00 mark on the 0° position). When the exact initial hand positions are found, the range capturing process starts and a first image is captured and sent to the RHM along with its associated information. Then, by controlling the elapsed time, several other pictures are taken at pre-defined time intervals. By controlling the elapsed time between each different image caught, the displacement of the hand is automatically known, as well as the value of the hand position angles. Given the fact that the background of a watch may contain diversified noise (logos, inscriptions, etc.), it is advisable to extract images from nearly all the background range, in order to allow the network to learn potential resemblances between certain background noise details and any of the hands. Moreover, the acquisition of this kind of information can be done from the planned set of training images with virtually no additional effort. For that, a portion of the background range should be obtained from the first image. In the last image the hands have moved to a different place from where they were in the first image, thus allowing the capture of the remaining background range. The collection of scaled rectangles from the obtained images are used to build the sets of examples for the training of the NN.

Considering the set of taken images, the sets used for the network training will include rectangles with the following contents:

- The H-hand and M-hand from all the images taken.
- Background noise from the first and last images.

The collection of all rectangles is then divided into two parts, respecting an equal distribution. One of the parts will constitute the TRAINING SET and the other the VALIDATION SET.

3 The training set

The choice of the range of images used for training the NN is described here. Before discussing the chosen range, the constraints inherent to the problem handled are elaborated upon, in order to explain the reasons of the decision taken.

3.1 Problems found

This section presents some particularities that appear as difficulties. They are presented below, as well as the solutions found to overcome or avoid them.

3.1.1 The date window

Certain watch types have a date window located in the centre-right part of the watch. In a first glance, the date window could be considered as normal background. The problem is that, normally, there is a magnifying glass above the date, integrated in the watch glass, that distorts the hands when positioned over the date region.

As depicted in figure 3, approximately half of the total area of a rectangle holding an M-hand positioned over the date window contains erroneous information. It is its rightmost part, corresponding to the outer part of the watch circle.

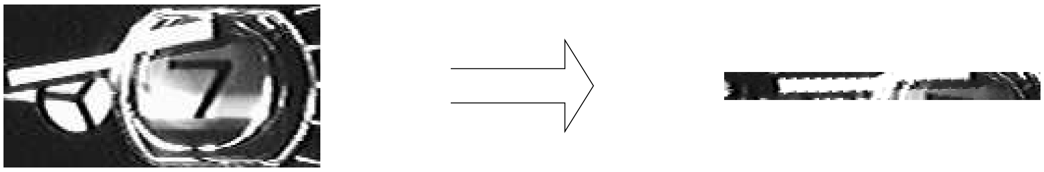

Figure 3: The M-hand over the date

Taking into account that a rectangle holding an H-hand will contain useless information in approximately the same portion of its area as the case above (see figure 4), it might be worth considering rejecting the rightmost half of the obtained rectangles, or at least giving it a lower importance, and perform the classification using mainly the information contained in the leftmost part, the one corresponding to the inner region of the circle. This might solve the problem of the date window and definitely avoid problems eventually caused by different background information existent in the rightmost part of an H-hand rectangle, depending on the region where that hand was positioned.

Figure 4: An example of an H-hand rectangle

The use of this technique would require, however, the leftmost part of the H-hand and the M-hand to be sufficiently distinguishable, which may not always be the case, as discussed below in section 3.1.2.

3.1.2 Difficult hand types

In an image recognition problem a NN is able to differentiate its patterns by learning the contents of the images. Hence, a problem may arise when a pattern contains no particularities, which is the case of plain hands with no internal structure or texture, and the only way of differentiating an H-hand from a M-hand being their different size.

Although this may not necessarily create a problem, it prevents the use of the technique consisting in not considering the rightmost half of the rectangle images, as proposed in the problem of the M-hand over the date window.

3.1.3 Superposing hands

One of the main difficulties in the project presented in this paper relates to when one of the hands is superposing the other, either partially or totally. The positioning of the hands in the hub follows the order, from bottom to top: HOURS, MINUTES, SECONDS. In certain types of watches, there is fourth hand, called the 24-HOURS hand, which makes one complete turn while the H-hand turns twice, that is, a turn a day. This hand, if existent, is located in the hub between the H-hand and the M-hand. Given this, three types of superposing are of interest:

The M-hand superposing the H-hand This case occurs often and deserves a special treatment. The network is able to solve the problem and recognize an H-hand when a M-hand is over it, except for a complete or almost complete superposing. However, in order to do it, the network has to be trained with such images.

The S-hand (or its tail) superposing the H-hand and/or the M-hand This kind of superposing occurs quite often and constitutes a great difficulty, specially with certain kinds of watches, where the size of the S-hand is approximately the same or bigger than the MINUTES hand. Additionally, the S-hand may extend back to the other side of the centre, producing a superposing in its angular position and another, partial, on the opposite angular position. The network is not able to solve any of these cases by itself without being trained for it. However, it becomes an awkward task to perform such training. So, the images received by the NN module have to be pre-selected. Since the ISM is capable of finding the position of the S-hand, it may reject images that present such problems before sending them to the next module. This problem only occurs in the recognition phase, since in the training phase the contents of the images are already controled.

The 24-HOURS hand superposing the H-hand As explained, the 24-HOURS hand, if existent, turns over the H-hand, which means that it may superpose it. Nevertheless, the training made with images where the M-hand superposes the H-hand prepares the network to work with these cases also, specially when the 24-HOURS hand is very thin, causing a small superposition, even when its angular position is the same as that of the H-hand.

3.2 The choice of the training images

Bearing in mind the constraints above, the training images should include the following two cases:

- The M-hand over the date window.
- The M-hand superposing the H-hand at diversified degrees.

Given this, the training time-range that was selected starts with the hands positioned at 2h07 and extends till they reach the time 2h14. It takes thus a little more than 7 minutes to capture the whole range. The shots are done at 1-minute intervals and as the S-hand should be prevented from superposing any of the other two, the images should be taken when it crosses the 12h00 zone, approximately. 8 images will so be obtained, but as the image corresponding to the 2h11 position has too high a rate of hands superposition, it is rejected, which leaves 7 images in total. It is important to remark that if a 24-HOURS hand exist, it should be positioned away from this range while the capturing takes place.

For the background images to be correctly captured, the watch should be positioned in a sufficiently correct position, meaning that the angular distance separating the 3h00 watch mark in the image from the 0° position should not exceed a few degrees (about 3°). Approximately half of the background patterns are extracted from image #1, the other half being extracted from image #7.

Table 1 describes the information extracted from each training image and the purpose given to it. $\varphi_H(0)$ and $\varphi_M(0)$ represent the angular positions of the H-hand and M-hand, respectively, at image #1.

Image #	Time position	Rectangles extracted: Angle	Rectangles extracted: Contents	Total	Destination
1	2h07	$\varphi_H(0)$	H-hand	1	Training Set (×4)
		$\varphi_M(0)$	M-hand	1	Training Set
		range of $\varphi \in [180°, 10°]$ in intervals of 5°	background	39	Distributed over both sets
2	2h08	$\varphi_H(0)$	H-hand	1	Validation Set (×4)
		$\varphi_M(0) - 6°$	M-hand	1	Validation Set
3	2h09	$\varphi_H(0) - 1°$	H-hand	1	Training Set (×4)
		$\varphi_M(0) - 12°$	M-hand	1	Training Set
4	2h10	$\varphi_H(0) - 1°$	H-hand	1	Validation Set (×4)
		$\varphi_M(0) - 18°$	M-hand	1	Validation Set
5	2h12	$\varphi_H(0) - 2°$	H-hand	1	Training Set (×4)
		$\varphi_M(0) - 24°$	M-hand	1	Training Set
6	2h13	$\varphi_H(0) - 3°$	H-hand	1	Validation Set (×4)
		$\varphi_M(0) - 30°$	M-hand	1	Validation Set
7	2h14	$\varphi_H(0) - 3°$	H-hand	1	Training Set (×4)
		$\varphi_M(0) - 36°$	M-hand	1	Training Set
		range of $\varphi \in [45°, 175°]$ in intervals of 5°	background	27	Distributed over both sets

Table 1: Rectangles extracted for training

The degree of superposition of the M-hand over the H-hand in image #6 is very high. So, the rectangles extracted from it go necessarily to the validation set to avoid teaching the network with examples of two different classes that resemble too much. The background range is divided into two parts with interleaved angular positions, one for the training set and the other for the validation set. Each H-hand rectangle is repeated 4 times in each of the sets, given that only a few images are used to form the sets and each acquired image of the H-hand is different from the other. The sets formed in this way will hold a total of 54 patterns for training and 47 patterns for validation. Table 2 shows the collection of rectangles with the H-hand and the M-hand used in the training set.

Image #	Rectangles extracted	
	H-hand	M-hand
1		
3		
5		
7		

Table 2: The H-hand and M-hand training rectangles.

4 The neural network configuration

Several different averaging factors were experimented to transform variable-sized rectangle images into fixed-sized ones. It was decided to set the rectangle images to a size of 100×10 pixels. This corresponds to an average one, considering the possible different existing watch sizes. The scaling is done in the RECTANGLE HANDLING MODULE, and the chosen fixed size gives rise to a neural network with an input layer size of 1000. Three classes are considered in this classification problem: HOURS, MINUTES, and NOISE. The three outputs are boolean.

The input and output signals of the network lay in the range [-1,+1]. So, all the 100×10 matrix grey-scale values of the rectangle images are scaled to fit this range. Given the fact that the network deals with a classification problem, it was decided to apply the *relative entropy* error measure, as proposed in [Hertz-91]:

$$E = \sum_{p=1}^{P} \sum_{j=1}^{N_L} \left[\frac{1}{2}(1+t_j) \log \left(\frac{1+t_j}{1+a_j} \right) + \frac{1}{2}(1-t_j) \log \left(\frac{1-t_j}{1-a_j} \right) \right].$$

t_j and a_j are the target and actual response values of output neuron j, N_L represents the number of output neurons, L the number of layers, and P the number of example patterns. The initial weights are set to random numbers in the range [-0.77,+0.77] [Thimm-94.2].

The value of the initial learning rate parameter is 0.3 and the momentum term is 0.9. The training is performed with the adaptive learning rate method proposed in [Silva-90]. This method was chosen from the set of adaptive learning rate methods described and implemented in [Moreira-95]. The choice was based both on its good performance results and low computational complexity as compared to the standard backpropagation algorithm. The principle used in the method is the following: Each connection weight has its own adaptive learning rate parameter that, at each iteration, is increased by a factor of 1.2 when the corresponding gradient component holds the same sign as in the previous iteration, being decreased by a factor of (1/1.2) otherwise. The backtracking procedure proposed by the authors that makes the algorithm return to the weight vector of the previous iteration when an increase of the error measure is detected, is not used, since it was observed to frequently cause convergence failures. Upper and lower bounds of 10^2 and 10^{-8}, respectively, were defined for the learning rate parameters. The algorithm uses batch updating of the weights and learning rates.

During classification, to select the output that is correct, the *winner takes all* method is used, that is, assuming the output value of +1 as True and -1 as False, the correct output signal, from all three, is considered to be the one that shows the value that is closest to +1.

The network is trained with the training set, and the validation set is presented after every two training epochs[1] to obtain a measure of the generalization ability. After the training has been completed, the set of weights chosen is the one that the network possessed during the epoch where the best generalization measure was obtained, that is, where the error measured for the validation set was the lowest.

It is very important that the light conditions used to capture the images for recognition are the same as used to obtain the images for the training. Otherwise, the levels of grey might be shifted, thus resulting in different input values to the network.

In a first stage, the neural network tests were being performed with three-layer[2] topologies. It was observed, however, that the existing three classes of this problem are linearly separable, which means that no hidden layer is needed in the network. This changes the scope of the neural network discussion to a different dimension - that of the perceptron. At least two consequences derive from this: the training is faster, because fewer weights are used and, in general, a perceptron takes very few epochs to be trained; the second consequence is that success is guaranteed to be obtained in the training due to the absence of local minima. But this also means that different training methods could be used - those that apply specially to the perceptron case.

In a linearly separable problem, and considering a weight space representation, the weight vector found as solution represents an hyper-plane separating the points of the two pattern classes. There are normally several solutions, each represented by a separating hyper-plane. The *algorithm of the optimal stability* [Krauth-87] finds a solution that maximizes the distance between the weight hyperplane and the points it separates. Its goal can be stated as finding the weight vector $\mathbf{w}$ that maximizes the expression

$$D(\mathbf{w}) = \min_p \ \mathbf{w}^T \cdot \mathbf{a}^p \, t^p,$$

where $\mathbf{a}^p$ and t^p are, respectively, the vector of input signals and the target output value for pattern p. An iterative version of this algorithm was implemented and used for the problem discussed here. It is described below:

0. $n \leftarrow 0$. Initialize and normalize the initial weight vector using the Euclidian norm: $\mathbf{w}(0) \leftarrow \frac{\mathbf{w}(0)}{\|\mathbf{w}(0)\|}$.

1. Find the set of S pattern points $k_1, ..., k_S$ located at distance $D(\mathbf{w}(n))$.

2. Build the correction vector $\mathbf{\Delta w}$ using the set of points k_s: $\mathbf{\Delta w} = \sum_{s=1}^{S} \mathbf{a}^{k_s} t^{k_s}$.

3. Find $\alpha > 0$ such that $D\left(\frac{\mathbf{w}(n)+\alpha\mathbf{\Delta w}}{\|\mathbf{w}(n)+\alpha\mathbf{\Delta w}\|}\right)$ is maximal.

4. If $\alpha = 0$ then stop.

5. Update the weight vector: $\mathbf{w}(n+1) \leftarrow \frac{\mathbf{w}(n)+\alpha\mathbf{\Delta w}}{\|\mathbf{w}(n)+\alpha\mathbf{\Delta w}\|}$.

6. $n \leftarrow n+1$. Go to 1.

[1] An epoch consists in the presentation of the complete set of training patterns.

[2] A network with three neuron layers is here considered as having one input, one hidden, and one output layer.

5 Results

Here are presented the neural network training results for the cases of the adaptive learning rate training algorithm and the optimal stability algorithm. It is shown that, although having different performances, neither of the two algorithms is capable of training the network for solving the problem completely. A new solution is then proposed and tested.

To measure the quality of the trained network, a TEST SET was propagated through it at the end of the training, using the optimal set of weights found. The contents of this set are different from any of the contents of the training and validation sets. It includes 118 rectangles of diversified background noise covering all the angular range, 47 H-hand rectangles at diversified angular positions, and 47 M-hand rectangles also taken from diversified angular positions, giving a total of 212 rectangles. The results presented in table 3 relate to simulations done with a series of ten experiments for each method, with different initial sets of weights. It shows the number of training epochs spent to obtain the optimal set of weights, the value of the error measure for each one of the training, validation and test sets for that set of weights, as well as the percentage of misclassified patterns in each of the three sets.

Method	Number of Epochs		Error Measure (Mean)			Percentage of Misclassification					
						Training		Validation		Test	
	Mean	σ	Train.	Valid.	Test	Mean	σ	Mean	σ	Mean	σ
Adaptive LR	10.0	1.3	0.000	2.145	76.439	0.0	0.0	0.0	0.0	1.6	0.7
Optimal Stability	15.8	4.1	47.429	47.714	258.138	16.5	25.7	18.1	23.1	16.8	14.4

Table 3: Results from the simulations.

The results of table 3 show a difference in the performance of the two methods. It can be observed that the method of the optimal stability does not achieve convergence and thus cannot be used. So, only the results of the the adaptive learning rate method will be discussed. Two things can be observed that derive from the fact that a linear problem is being handled and that a perceptron is thus applied, instead of a multilayer network: the training is completed in a small number of iterations and the error in the training set always goes down to a value of zero. Nevertheless, some misclassifications are still observed in the test set. With an analysis of the kind of misclassifications occurred, three different problems were found. The rectangles relating to each of them are shown in table 4.

Problem	Rectangles
1	
2	
3	

Table 4: The misclassified rectangles.

The first problem relates to the S-hand superposing the H-hand and the M-hand. As mentioned before, no training is done with this kind of superposing, which means that situations like these are to be expected and avoided during the processing sequence before the network has to deal with them. So, although it remains a problem, it is supposed to be solved by a processing module other than the NNM.

The second problem is critical. The rectangle of table 4 shows an H-hand positioned over the date region. Up till now, only the problem of the M-hand over the date has been considered, but another obstacle appears caused by the fact that the network is trained with no H-hand rectangles over the date region, as the rectangles in table 2 illustrate. If the range selected for the capturing of the training images includes the H-hand over the date region, the outcome is that the network will be unable to recognize this hand when it is outside the date. A capturing range which includes both kinds of situation would be too long to obtain, due to the speed of the H-hand displacement. Furthermore, a second operator-based watch setting is to be avoided, as already discussed. So, the problem is supposed to be solved by pure neural network performance and capability of recognizing somewhat different objects of the same class, even when trained with only one of them. It may be concluded, then, from the performance results of table 3 that the neural network settings used are not optimized.

The same conclusion can be derived from the third problem. Both rectangles contain a 24-HOURS hand. In the upper rectangle, it prevents the network from classifying the rectangle as an H-hand, as it would be desired. In the lower rectangle it is simply confused with an M-hand. It should be noted that although the 24-HOURS hand is included in the training, it is not captured explicitely as a different hand - it is instead captured as background noise, being located somewhere in the obtained background range.

The solution to the problem was searched in the perceptron theory, and a very simple principle was found and applied to modify the network settings: the initial weight vector was set to zero. Table 5 shows the results for the simulation done for each method using this feature.

Method	Number of	Error Measure			Percentage of Misclassification		
	Epochs	Training	Validation	Test	Training	Validation	Test
Adaptive LR	6	0.0000	0.1138	19.0959	0.0000	0.0000	0.9434
Optimal Stability	10	0.1812	1.5067	29.8125	0.0000	0.0000	0.9434

Table 5: Results from the simulations with $\mathbf{w}(0) = \vec{\mathbf{0}}$.

Although the difference in the percentage of the test set misclassification is not considerable as compared to table 3, there is a big difference in the test set error measure. In any case, the performance of the network, relating to generalization, is substantially better with the new setting. With an analysis of the misclassified patterns, it is observed that problems 2 and 3 of table 4 are now solved, with problem 1 persisting, which was to be expected.

An important result concerns the fact that the algorithm of the optimal stability is now able to find a valid solution. Comparing the two methods, they show slightly different error measures, although the final generalization performance is similar. The greatest difference lies in the time spent on training. The number of iterations is smaller for the adaptive learning rate method. This is enhanced by the difference in complexity per iteration, with the algorithm of the optimal stability being considerably more computationally intensive, caused by the search for the parameter α which demands several training set presentations per iteration.

6 Conclusions

An overview of a Neural Network Module module integrated in a watch Quality Control System is herein presented. The processing sequence used by the system to automatically determine the time is described. It is then explained how to train the NNM as well as how to gather the training information, taking into account the inherent constraints of the problem. Details such as the magnifying glass over the date region that distorts the hands located there, hands that are partially superposed by other hands, and the existence of H-hands and M-hands whose only difference is their size, increase the difficulty of the problem. It is explained how to obtain a restricted set of training images, whose acquisition does not result in overcharged time and effort in the training preparation, but that at the same time allows the network to solve the above problems. A description of the neural network settings used is given and two different training algorithms are presented, which includes one with adaptive learning rate and another that performs optimization of a linear problem, applied to perceptrons, given that the problem is linearly separable. Results of experiments performed with both methods are presented, from which it may be concluded that the method of adaptive learning rate is preferable, both on training speed and quality of the solution.

Given that the Neural Network Module described here is in the prototype phase, further improvements and adjustments are planned, particularly during the integration phase of all the modules to create the final system. A valid solution is already proposed, however, one with which it is possible to solve the problem.

The tests done here are based on one type of watch. Although this watch type includes the most important difficulties to handle in this problem, further tests have to be done with other watch types to confirm the obtained results and introduce adjustments, if necessary.

Acknowledgements

The authors would like to thank Gilbert Maitre, Eddy Mayoraz and Perry Moerland for the assistance provided.

References

[Hertz-91] John A. Hertz, Anders S. Krogh, and Richard G. Palmer. "*Introduction to the Theory of Neural Computation*" (ISBN: 0-201-50395-6), volume I of *Computation and Neural Systems Series; Santa Fe Institute Studies in the Sciences of Complexity; Lecture notes.* Addison-Wesley Publishing Company, The Advanced Book Program, Redwood City, California, 1991.

[Krauth-87] Werner Krauth and Marc Mézard. "Learning Algorithms with Optimal Stability in Neural Networks." *Journal of Physics A: Mathematical and General* (ISSN: 0305-4470), volume 20, pages 745–752. Institute of Physics, London, 1987.

[Moreira-95] Miguel Moreira and Emile Fiesler. "Neural Networks with Adaptive Learning Rate and Momentum Terms." Technical Report 95-04, IDIAP, Martigny - Switzerland, October 1995.

[Silva-90] Fernando M. Silva and Luis B. Almeida. "Speeding Up Backpropagation." In Rolf Eckmiller (editor), *Advanced Neural Computers ((Proceedings of the) International Symposium on Neural Networks for Sensory and Motor Systems (NSMS))* (Neuss, Germany; March 22–24, 1990) (ISBN: 0-444-88400-9 (U.S.); LCCCN: QA76.87.A37), pages 151–158. Elsevier Science Publishers B. V. (North-Holland), Amsterdam, The Netherlands, 1990.

[Thimm-94.2] G. Thimm and E. Fiesler. "Weight Initialization for High Order and Multilayer Perceptrons." In Marc Aguilar (editor), *Proceedings of the '94 SIPAR-Workshop on "Parallel and Distributed Computing"* (Institute of Informatics, University of Fribourg; Fribourg, Switzerland), pages 87–90. Institute of Informatics, University of Fribourg, Fribourg, Switzerland, 1994.