András GYÖRGY, Tamás BARBARICS, József PADÁNYI

LANDMINE LOCALIZATION USING NEURAL NETWORK

The purpose of the research is to develop a method that can make minesweeping safer by no longer demanding human assistance for the localization of landmines. The objective is to create a robot that can find specific landmine types by their geometric features. For this, the required knowledge is acquired by methods based on artificial neural networks. The final result is the demonstration of the algorithm of an intelligent robot's pattern recognition module that can achieve the first step of demining an area, which is localizing mines without risking human life.

INTRODUCTION

One of the main imperfections of landmines is that the target can not be chosen, so the resident population, workers of humanitarian associations and the natural wildlife are also in great danger just like soldiers. There are only estimations about the number of accidents, since most of them take place in developing countries, but approximately 20.000 tragedies happen each and every year, which means two per hour (for example, between 1979 and 2005 in Cambodia more than 45.000 people were injured, 75% of them was civilian[1]).

There are more than 80 countries all over the world known as minefields, and no one knows the exact number and position of mines about a certain area. The most dangerous places are Afghanistan, Angola, Burundi, Bosnia and Herzegovina, Cambodia, Chechnya, Colombia, Iraq, Nepal and Sri Lanka. In addition to this, some of the countries do not provide official and precise report concerning the problem, like Burma, India and Pakistan [1].

Several demining methods have been developed [2], of which the most important are manual search with metal detector, trained dogs, bees[3] , rodents[4] , plants [5], bacteria [6], nuclear and acoustic techniques [7].

Every method has its own disadvantages: there are always types of landmines that can not be detected with the given technique, like plastic mines when using metal detector. Another aspect is when we use animals in order to acquire information, we place another living being in great danger, which is highly morally questionable, not to mention that the biology of animals can hugely affect the indication result.

MAIN CONCEPT

Regarding the fact, that landmines have well–defined geometry, we can assume that a novel technique based on geometric features can be very successful in the process of demining. Although, since the orientation vary in great range, and the environment is rather noisy, a hard computing pattern recognition method would not be flexible and robust enough, so we suggest a soft computing approach, that is based on artificial neural networks.

The general modus operandi is the following (figure 1): the output (y) is typically a nonlinear mapping of the input vector (x). The expected operation is reached by teaching the neural network based on its error (ε), which can be derived from the desired outputs (d) for the learning samples.

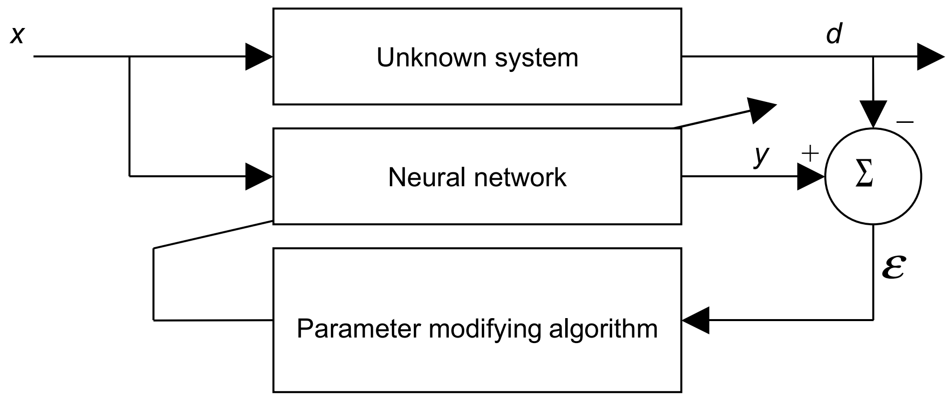

Figure 1: Structure of the learning machine

Our goal is to develop such an algorithm, so the demining process would risk human life no longer. The allowed error is asymmetric, since false positive signals are much less problematic than false negative ones, which would mean we missed a real mine. At the evaluation of results this will be a major aspect.

Considering the required data, the tasks to be fulfilled are the following ones:

- acquiring data;
- create database (human knowledge);
- choose the structure of neural network (by a priori knowledge);
- teaching;
- testing.

Since we do not have the real working machine, we do not possess real samples. It follows that we have to generate them by computer paying attention to reality, so a great amount of noise has to be added. By this we can claim that the generated samples are almost like the real ones.

LEVENBERG–MARQUARDT METHOD

There are many ways to teach a neural network, but in this case the best performance was reached by the widely used Levenberg–Marquardt method . It drastically reduces the number of training epochs at the price of enhanced computational capacity requirement.

Gradient based methods use the $\underline{\underline{R}}$ autocorrelation matrix to gain additional information about the error surface. In case of quadratic error surface, the $\underline{\underline{H}}$ Hesse–matrix is equal to $\underline{\underline{R}}$, so it provides information about the curvature. The main advantage of this approach is that the inverse of $\underline{\underline{R}}$ is not required, which is a great computational saving.

In case on non–quadratic error surface, it can be approximated by linearization, so the $C(\underline{w})$ error function can be estimated by the first three elements of its Taylor–polynom:

$$C(\underline{w}) \simeq C(\underline{w}[k]) + \nabla C(\underline{w}[k])^T (\underline{w} - \underline{w}[k]) + \frac{1}{2}(\underline{w} - \underline{w}[k])^T \underline{\underline{H}} (\underline{w} - \underline{w}[k]) \,, \tag{1}$$

where

$$\underline{\underline{H}} = \frac{\partial \nabla C(\underline{w}[k])}{\partial \underline{w}} \,. \tag{2}$$

The location of the minima can be determined by derivation of the expression by $\underline{w}$:

$$\nabla C(\underline{w}[k+1]) = \nabla C(\underline{w}[k]) + \underline{\underline{H}}(\underline{w}[k])(\underline{w}[k+1] - \underline{w}[k]) = \underline{0} \,, \tag{3}$$

so the learning rule is the following:

$$\underline{w}[k+1] = \underline{w}[k] - \underline{\underline{H}}(\underline{w}[k])^{-1} \nabla C(\underline{w}[k]) \,. \tag{4}$$

The proposition of the Levenberg–Marquardt method is a mixture, so the weight modification can be computed by:

$$\underline{w}[k+1]=\underline{w}[k]-\left[\underline{\underline{H}}(\underline{w}[k])+\lambda[k]\underline{\underline{I}}\right]^{-1}\underline{\nabla}C(\underline{w}[k]) \ . \tag{5}$$

Coefficient $\lambda[k]$ is step–dependant and it defines the ratio of the steepest descent and the Newton–method: if little, then the first one dominates, otherwise the second one. It can be regarded as a kind of regularization that helps when the Hesse–matrix is singular or close to it.

Additional computation reduction can be achieved if we have quadratic error function. In this case

$$C(\underline{w})=\frac{1}{2}E\left\{(d-y(\underline{w}))^2\right\}=\frac{1}{2}E\left\{\varepsilon^2\right\} \ . \tag{6}$$

so the gradient vector is

$$\underline{\nabla}C(\underline{w})=-E\left\{\varepsilon\underline{\nabla}y(\underline{w})\right\} , \tag{7}$$

and the Hesse–matrix is

$$\underline{\underline{H}}=E\left\{\underline{\nabla}y(\underline{w})\underline{\nabla}y(\underline{w})^T-\varepsilon\left[\frac{\partial^2 y(\underline{w})}{\partial w_i\partial w_j}\right]\right\} \ . \tag{8}$$

Approaching the optima the error is decreasing, so the second element can be neglected. By this simplification, we do not need the second partial derivatives only the first ones to approximate the Hesse–matrix.

DETECTION

The major component of the detector is a simple piston (figure 2) with a valve equipped with a metal pin with which we can gain information about the possible vertical position of the landmine in the ground (*L*): the greater the shift the lower the mine is.

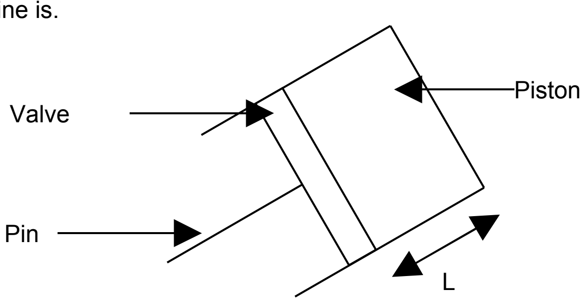

Figure 2: Detector

If we place 20 of these simple indicators in a line with a distance of five centimeters, we can obtain information of a one meter long line, so if we execute 20 measurements, we know the required data of one square meter. It is noticeable that the data acquisition is very simple, fast and does not require human assistance, which is a great advantage of the suggested method.

Another feature is that on different land types there is only one parameter that we have to set: the pressure of the piston. In this way, the method can be easily used in a Cambodian forest as well as in an Iraq desert.

MAIN PATTERNS

If we observe the landmine types that are used in the last decades (figure 3) we can notice a typical feature: they are rather circular in a horizontal segment.

Figure 3: Typical landmine forms

To demonstrate that the suggested method is flexible and can be easily used for different types of landmines, we will search for two different landmines. Both of them are of 20–25 cm diameter, one of them is a simple disc, the other has a knob in the center. If we use a data acquisition grid with 5 cm segments, the observed landmine types can be characterized with a 5×5 square and 13 measured inputs (figure 4).

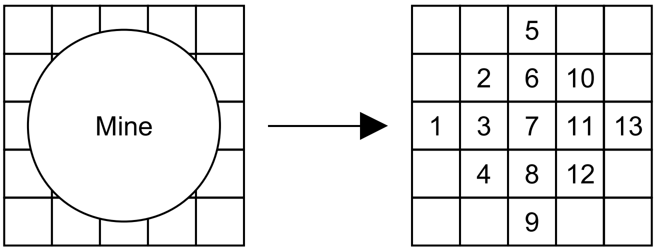

Figure 4: Mapping

In this way, our „virtual landmines" (figure 5) can be seen as vectors of 13 elements, where the measured value (deepness) can be from the set of {0;1;2}:

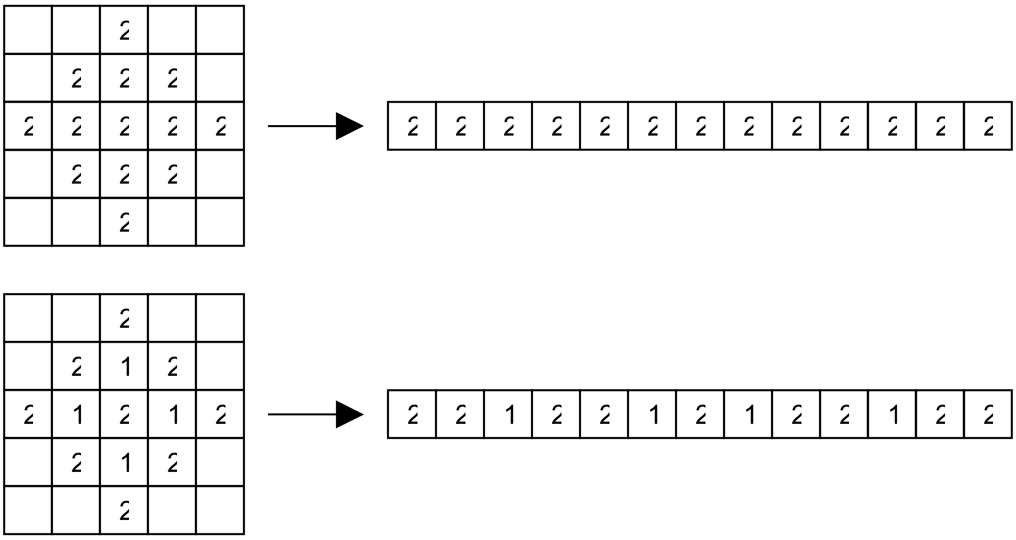

Figure 5: Landmine type I. and II.

In practice, landmines are positioned by airplanes or artillery, so their lay is not perfectly horizontal. In addition to this, we are not aware of which side is in front of us, so we have to pay attention to this invariance (skewness and orientation). As a result, we have eight other main patterns for each type in addition to the mentioned ones (figures 6 and 7):

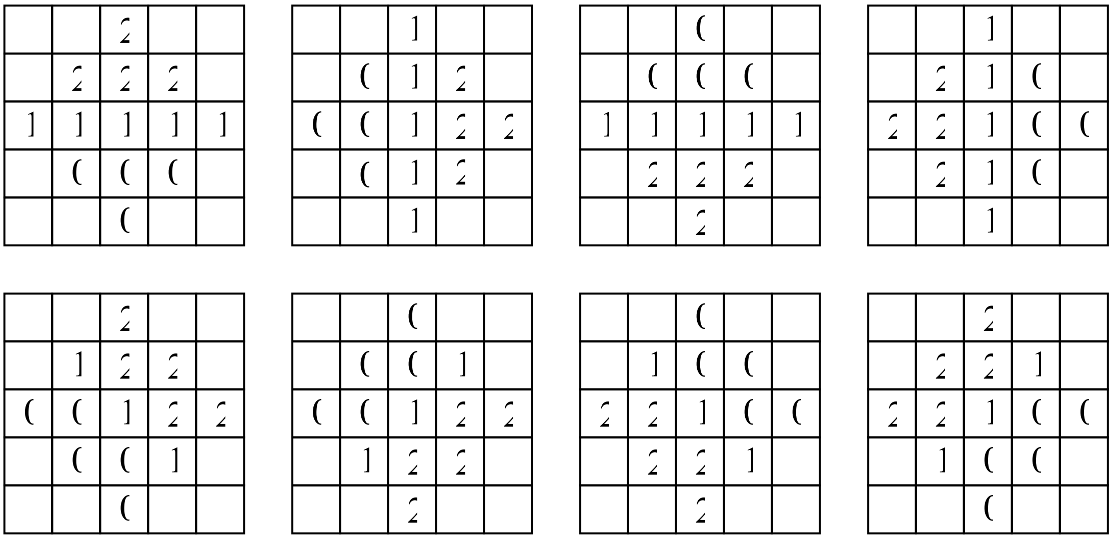

Figure 6: Main patterns for landmine type I.

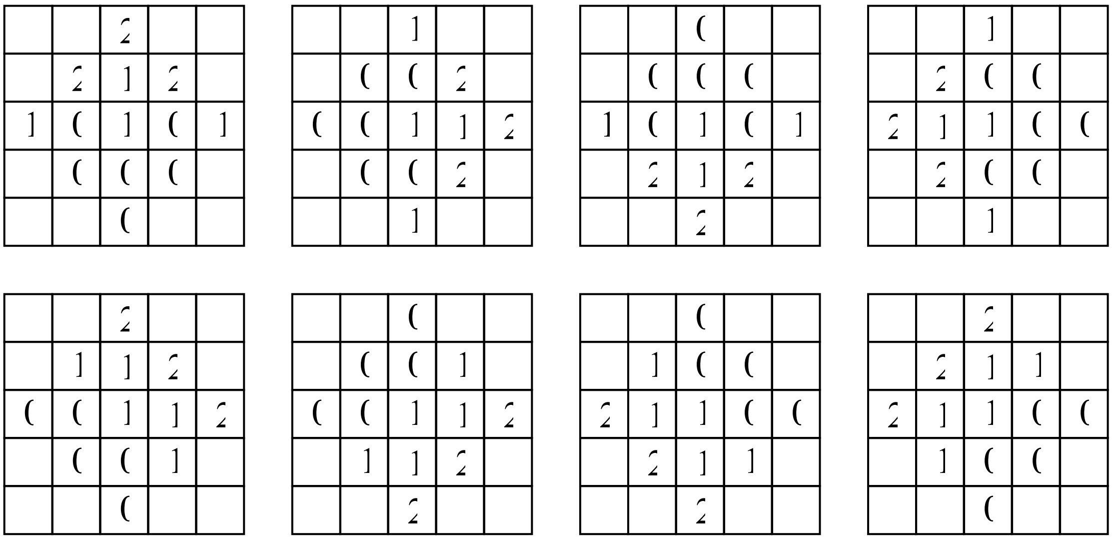

Figure 7: Main patterns for landmine type II.

DECOMPOSITION

If we simply used these vectors as the input of the neural network, the size of it would be huge and the learning speed would be very small. Instead of this, we should decompose the pattern to smaller ones, so that the dimension of the input space decreases. With this simplification, the learning speed can be hugely increased.

Concerning the main patterns, we can divide them into smaller groups, the so–called partial patterns (tables 1 and 2). It is noticeable, that we reduced the input dimension to 3–5 from 13, which is a remarkable simplification.

	1	2	3	4	5	6	7	8	9	10	11	12	13
#1	×	2	×	×	2	2	×	×	×	2	×	×	×
#2	×	×	×	×	×	×	×	×	×	2	2	2	2
#3	×	×	×	2	×	×	×	2	2	×	×	2	×
#4	2	2	2	2	×	×	×	×	×	×	×	×	×
#5	×	0	×	×	0	0	×	×	×	0	×	×	×
#6	×	×	×	×	×	×	×	×	×	0	0	0	0
#7	×	×	×	0	×	×	×	0	0	×	×	0	×
#8	0	0	0	0	×	×	×	×	×	×	×	×	×
#9	2	2	2	×	2	2	×	×	×	×	×	×	×
#10	×	×	×	×	2	2	×	×	×	2	2	×	2

#11	×	×	×	×	×	×	×	2	2	×	2	2	2
#12	2	×	2	2	×	×	×	2	2	×	×	×	×
#13	0	0	0	×	0	0	×	×	×	×	×	×	×
#14	×	×	×	×	0	0	×	×	×	0	0	×	0
#15	×	×	×	×	×	×	×	0	0	×	0	0	0
#16	0	×	0	0	×	×	×	0	0	×	×	×	×
#17	1	×	1	×	×	×	1	×	×	×	1	×	1
#18	×	×	×	×	1	1	1	1	1	×	×	×	×
#19	2	×	2	×	×	×	2	×	×	×	2	×	2
#20	×	×	×	×	2	2	2	2	2	×	×	×	×
#21	×	×	×	1	×	×	1	×	×	1	×	×	×
#22	×	1	×	×	×	×	1	×	×	×	×	1	×

Table 1: Partial patterns for landmine type I.

	1	2	3	4	5	6	7	8	9	10	11	12	13
#1	×	2	×	×	2	1	×	×	×	2	×	×	×
#2	×	×	×	×	×	×	×	×	×	2	1	2	2
#3	×	×	×	2	×	×	×	1	2	×	×	2	×
#4	2	2	1	2	×	×	×	×	×	×	×	×	×
#5	×	0	×	×	0	0	×	×	×	0	×	×	×
#6	×	×	×	×	×	×	×	×	×	0	0	0	0
#7	×	×	×	0	×	×	×	0	0	×	×	0	×
#8	0	0	0	0	×	×	×	×	×	×	×	×	×
#9	2	2	×	×	2	×	×	×	×	×	×	×	×
#10	×	×	×	×	2	×	×	×	×	2	×	×	2
#11	×	×	×	×	×	×	×	×	2	×	×	2	2
#12	2	×	×	2	×	×	×	×	2	×	×	×	×
#13	0	0	0	×	0	0	×	×	×	×	×	×	×
#14	×	×	×	×	0	0	×	×	×	0	0	×	0
#15	×	×	×	×	×	×	×	0	0	×	0	0	0
#16	0	×	0	0	×	×	×	0	0	×	×	×	×
#17	×	1	×	×	×	1	1	×	×	×	1	1	×
#18	×	×	×	1	×	×	1	1	×	1	1	×	×
#19	×	1	1	×	×	×	1	1	×	×	×	1	×
#20	×	×	1	1	×	1	1	×	×	×	1	×	×
#21	1	×	0	×	×	×	1	×	×	×	0	×	1
#22	×	×	×	×	1	0	1	0	1	×	×	×	×
#23	2	×	1	×	×	×	2	×	×	×	1	×	2
#24	×	×	×	×	2	1	2	1	2	×	×	×	×

Table 2: Partial patterns for landmine type II.

PREPROCESSING

We only discussed the noise–free case so far, but in reality there is always a great amount of disturbing noise, so the patterns to recognize are definitely not such easily identifiable. We presented the main and the partial patterns, but only with discrete numbers (from the set of {0;1;2}). In practical use, the measured input is from the continuous range of [0,2]. If we simply use the raw values, the learning takes too long and the distinction is not contenting enough. The other extremity is naively rounding them, but this is not robust enough, because it is very sensitive to noise. The suggested preprocessing method is somewhere in the middle.

We divide the range to discrete domains that are continuous, so we gain the advantages of the two approaches, but throw away their drawbacks. With the discrete domains we can roughly characterize the deepness to „LOW", „MEDIUM" and „HIGH", so the separation is satisfying and the learning is fast, and with the continuity of the domains we can distinguish the fine features, so the method is tolerant to noise.

$$\hat{x}_{preproc} = \begin{cases} c \cdot x^2 & , \quad if \quad x \in [0;0.5) \\ 1 + \operatorname{sgn}(x-1) \cdot c \cdot (x-1)^2 & , \quad if \quad x \in [0.5;1.5) \\ 2 - c \cdot (x-2)^2 & , \quad if \quad x \in [1.5;2] \end{cases} . \tag{9}$$

Since the activation function of neurons is steep in the neighborhood of zero, we should shift and narrow the region:

$$x_{preproc} = \frac{\hat{x}_{preproc} - 1.5}{2} . \tag{10}$$

TOPOLOGY

Regarding theoretical results, we used Multi Layer Perceptrons to fulfill the classifications. In order to decrease the number of neurons and the learning time, we took advantage of our priori knowledge: we created subnets to learn specific subproblems, so we constructed a hierarchical structure (figure 8).

There are three layers of subnets: in the first one we can find 22 or 24 subnets depending on the landmine type: their objectives are identifying the partial patterns. The second layer executes the composition: it learns the appropriate coexistence of the specific partial patterns, so it detects the main patterns. The task of the third layer is to summarize the knowledge, whether the input vector is from a landmine or not.

Since the required knowledge for us is a „yes or no", we need a stepfunction at the end of the third layer, whose comparation level is noise–dependent.

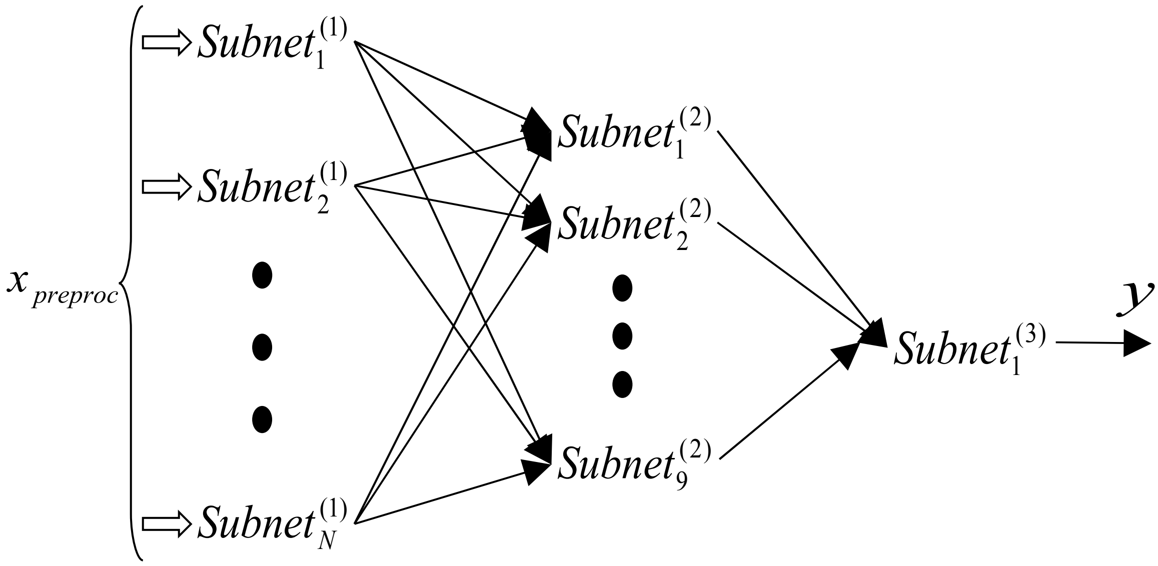

Figure 8: Topology multimodular MLP

DATA SET

Usually we divide our data set to three disjunct sets: learning, validation and test set. The first one is for teaching the network, the second is to check the performance during the teaching (and early stopping), and the last one is for the ultimate test after teaching.

Since in our case the precision and the knowledge of precision are essential, we separate a test set before the learning phase, so we can only define the ratio of the learning and validation set sizes.

To get relevant and objective information about the performance, we place „almost mines" in the minefield when we test our system, so we can decide whether we can identify real mines and neglect everything else or not.

GRAPHICAL USER INTERFACE

A special Graphical User Interface (figure 9) is very helpful, because we can easily set and modify the most important parameters, like landmine type, teaching and network parameters, set sizes, number of landmines and the amount of noise.

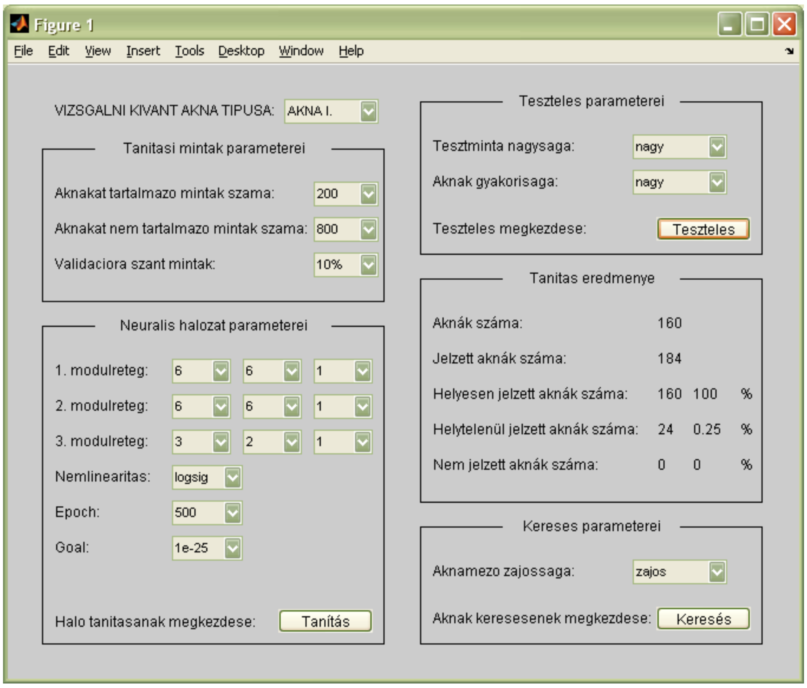

Figure 9: Graphical User Interface

Another very useful application is the Result Window that provides numerical and graphical information as well: we gain knowledge on the performance and see the observed minefield with the found and the missed landmines.

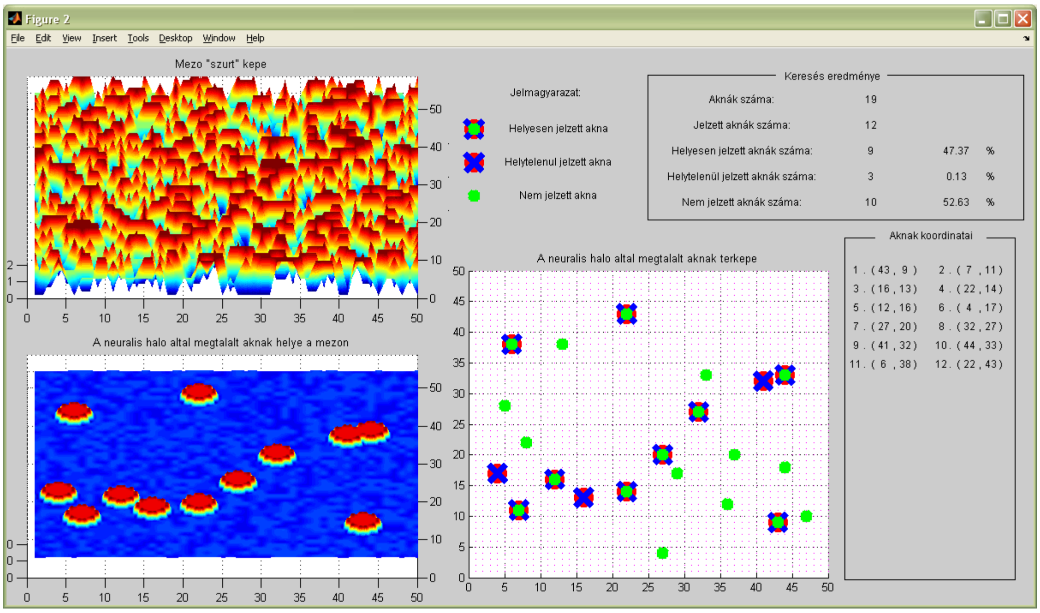

Figure 10: Quantitative and graphical results

TEACHING RESULTS

We can see the typical teaching phase on figure 11: performance is getting better rapidly for a few dozens of epochs, after that we notice stagnation, so it is practical to stop teaching (early stopping).

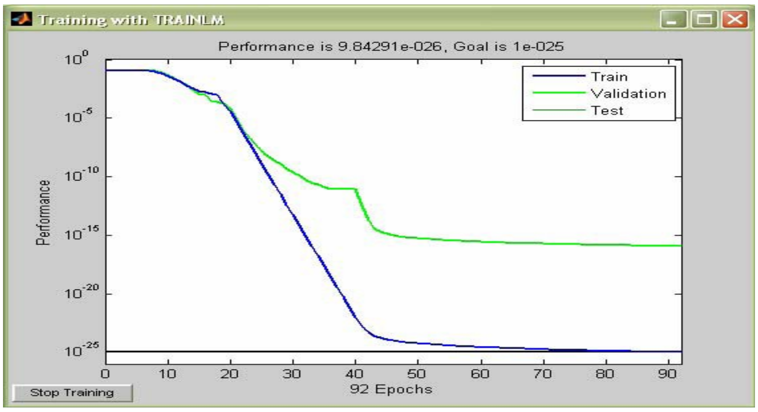

Figure 11: Typical teaching phase

In tables 3 and 4 we can see the teaching results for both landmine types. The observed network sizes are „too small”, „appropriate” and „too large”, and the most important numerical results are „correct”, „false positive” and „false negative” identifications.

Landmine Type I.	Small network Layer 1: 3-3-1 neurons Layer 2: 3-3-1 neurons Layer 3: 3-3-1 neurons		Appropriate network Layer 1: 6-6-1 neurons Layer 2: 6-6-1 neurons Layer 3: 3-2-1 neurons		Large network Layer 1: 10-10-1 neurons Layer 2: 10-10-1 neurons Layer 3: 5-5-1 neurons	
Correct Identifica-tions	**90**	**56.60%**	**160**	**100.00%**	**155**	**100.00%**
False Positives	**6**	**0.06%**	**24**	**0.25%**	**19**	**0.2%**
False Neg-atives	**69**	**43.40%**	**0**	**0.00%**	**0**	**0.00%**

Table 3: Teaching results of landmine type I.

Landmine Type II.	Small network Layer 1: 3-3-1 neuron Layer 2: 3-3-1 neuron Layer 3: 3-3-1 neuron		Appropriate network Layer 1: 7-7-1 neuron Layer 2: 7-7-1 neuron Layer 3: 4-3-1 neuron		Large network Layer 1: 10-10-1 neuron Layer 2: 10-10-1 neuron Layer 3: 5-5-1 neuron	
Correct Identifica-tions	**71**	**47.33%**	**135**	**100.00%**	**151**	**100.00%**
False Positives	**32**	**0.34%**	**18**	**0.19%**	**15**	**0.16%**
False Neg-atives	**79**	**52.67%**	**0**	**0.00%**	**0**	**0.00%**

Table 4: Teaching results of landmine type II.

Looking at the results we can reveal that there is no great difference between the two landmine types and the performance of the network of appropriate size is satisfying.

TEST RESULTS

In the final test phase, we created a completely new environment: there are not only virtual mines in the observed minefield but „almost mines” as well, as mentioned before. Without this complication we could not be sure about the performance of the system, because in a real minefield there are branches, bones and rocks that have to be separated from searched mines.

In tables 5 and 6 we provide information about test result: the fields are the same as before, and it is noticeable that the numerical results are only slightly different.

Landmine Type I.	Small network Layer 1: 3-3-1 neurons Layer 2: 3-3-1 neurons Layer 3: 3-3-1 neurons		Appropriate network Layer 1: 6-6-1 neurons Layer 2: 6-6-1 neurons Layer 3: 3-2-1 neurons		Large network Layer 1: 10-10-1 neurons Layer 2: 10-10-1 neurons Layer 3: 5-5-1 neurons	
Correct Identifica-tions	**9**	**47.37%**	**18**	**100.00%**	**18**	**100.00%**
False Positives	**3**	**0.13%**	**1**	**0.04%**	**1**	**0.04%**
False Neg-atives	**10**	**52.63%**	**0**	**0.00%**	**0**	**0.00%**

Table 5: Test results of landmine type I.

Landmine Type II.	Small network Layer 1: 3-3-1 neuron Layer 2: 3-3-1 neuron Layer 3: 3-3-1 neuron		Appropriate network Layer 1: 7-7-1 neuron Layer 2: 7-7-1 neuron Layer 3: 4-3-1 neuron		Large network Layer 1: 10-10-1 neuron Layer 2: 10-10-1 neuron Layer 3: 5-5-1 neuron	
Correct Identifica-tions	**9**	**45.00%**	**18**	**100.00%**	**18**	**100.00%**
False Positives	**1**	**0.04%**	**5**	**0.22%**	**2**	**0.09%**
False Neg-atives	**11**	**55.00%**	**0**	**0.00%**	**0**	**0.00%**

Table 6: Test results of landmine type II.

With the small network about half of the landmines are correctly detected and there are only a few false positive signals. If we increase the number of neurons to the appropriate level, in other words we increase the complexity the performance is getting better: the number of correct identifications increases and the prevalence of false negatives decreases. With additional neurons we do not reach remarkable improvement, only the learning speed increases.

We have to emphasize that the comparison level of the last layer is set to the network of appropriate size, so the probability of false negative signals could be cut back at a cost of increasing the number of false positive identifications.

PRACTICAL ASPECTS

The demining robot can be naturally divided to thee major components: the moving platform, the sensors and the algorithm. The first two hardware type machines can be constructed on a very low budget, since high precision is not required, which was the financial objective. The last element, the neural network itself can be implemented on a microcontroller, so it is also a low–cost solution. Because of reasonable costs, the development can be commenced easily.

CONCLUSION

To summarize the achieved results, we can claim that a soft computing based demining method was presented with great performance in a virtual minefield. The algorithm is very flexible and robust, thanks to the neural network approach. The professional objective is fulfilled, since the ratio of correct identifications is very high, and the false negative indications are rare.

Keywords: Soft computing, artificial neural network, demining

REFERENCES

[1] http://www.icbl.org/

[2] http://en.wikipedia.org/wiki/Demining

[3] Bromenshenk, J.J., (et al.): "Bees used in Area Reduction and Mine Detection". -in.: Journal of Mine Action, Issue 7.3., 2003.

[4] http://www.aegis.com/news/re/2004/RE040956.html

[5] http://www.acfnewsource.org/science/mine_sniffing_plants.html

[6] http://www.mech.uwa.edu.au/jpt/demining/others/ornl/rsb.html

[7] http://technology.newscientist.com/article/dn10868-vibrations-could-reveal-landmine-locations.html

[8] ALTRICHTER, M., HORVÁTH G., PATAKI B., STRAUSZ Gy., TAKÁCS G., VALYON, J.: Neurális hálózatok. -Bp.:Panem Kiadó, 2006.