COIN SIGNAL RECOGNITION BASED ON NEURAL NETWORKS

Cristian MONEA[1], Gabriel V. IANA[2]
[1]University of Pitesti, Romania
[2]University of Pitesti, Romania
[1]cristi_m90@yahoo.com, [2]gabriel.iana@upit.ro

Keywords: neural network, acoustic fingerprinting, natural frequencies, example-based learning, MATLAB GUI

Abstract: *The purpose of this paper is to develop a method for identifying coins by the sound generated when they hit a hard surface, using a neural network. Generally, coin identification in vending machines, for e.g., is done using magnetic or optical methods. This paper focuses on the acoustic method, in which coin recognition is based on the detection of the coin's natural frequencies. The frequencies of these vibrations depend on the object's properties (mass, shape, material type), and remain the same as long as these properties do not change, thus being used as acoustic fingerprints. Also, this method permits recognition of fake or deteriorated coins, because they have different properties. The principle applied in this paper can be used for the recognition of numerical sequences produced by other objects.*

1. INTRODUCTION

Artificial intelligence is a branch of computer science that studies and develops programs and systems which solve complex problems using qualitative or intuitive methods, similar to human functions, instead of quantitative methods and calculations based on accurate data, specific to conventional computing systems. The development of this research field has allowed, for example, the development of systems controlled by fuzzy logic or neural networks. The example-based learning and pattern recognition properties of neural networks make possible their use in coin recognition applications, such as vending machines.

Generally, coin identification in vending machines, for e.g., is done using magnetic or optical methods. This paper focuses on the acoustic method, in which coins are identified by their acoustic fingerprints, generated when they hit a hard surface. The frequencies of these vibrations depend on the coin's physical properties: mass, shape, dimensions, material type, elastic modulus, and remain the same as long as these properties do not change [1].

This paper combines the flexibility offered by the neural network with the principle of acoustic fingerprinting of objects to create an algorithm which can learn a set of numerical sequences, the sounds generated by the coins which will be identified, and permit the recognition of these sequences.

2. THEORETICAL BACKGROUND

An essential part of the algorithm is the neural network. Neural networks are systems which model the functioning of biological nervous systems. Their main feature, which differentiates them from fuzzy or expert systems, is the ability to learn from examples [2].

The history of neural networks can be structured in three periods: academic, expansion and production (Fig. 1) [2].

In 1943 Warren S. McCulloch and Walter H. Pitts proposed the first artificial neuron model. After slowly losing attention, neural networks became a topic of interest in 1986, when David Rumelhart and James McClelland proposed the backpropagation algorithm as a training method. The production period starts with the publication of the first neural networks journal, *IEEE Transactions on Neural Networks*.

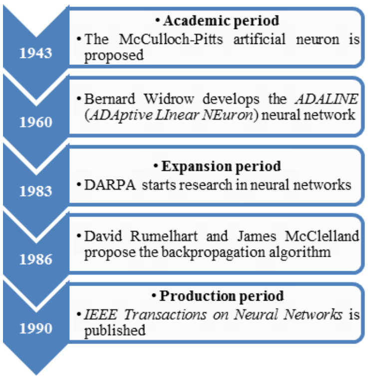


Fig. 1 Brief history of neural networks

The building block of a neural network is the artificial neuron. It models the functioning of a biological neuron through its multiple inputs, weights and transfer function. Fig. 2 shows a McCulloch-Pitts artificial neuron.

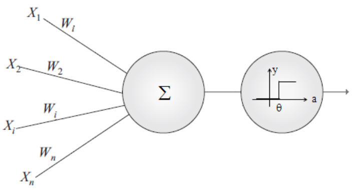


Fig. 2 The McCulloch-Pitts artificial neuron

There are many types of transfer functions. A common transfer function is the unipolar sigmoid (Fig. 3). This transfer function is often used because it models more realistic the biological neuron [3].

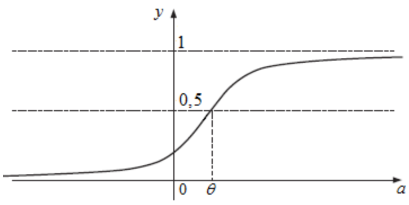


Fig. 3 Unipolar sigmoid transfer function

The neuron response is equal to the weighted sum of the inputs. In the case of a step transfer function, if this sum is greater than a threshold, then the neuron outputs 1. Otherwise, it outputs 0. In the case of a unipolar sigmoid transfer function, the neuron's output is a value in the (0,1) interval.

Multilayer perceptron networks are sets of artificial neurons grouped in three or more layers: input, hidden (at least one layer) and output (Fig. 4). Each neuron in one layer has connections with every neuron in the next layer. The information is scattered throughout the network, being stored in its weights.

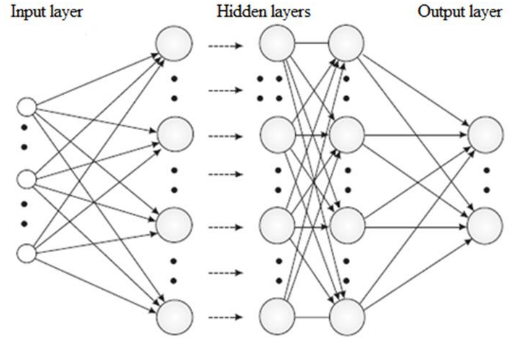


Fig. 4 MLP neural network

The backpropagation algorithm is used to train the network (Fig. 5). If convergence is achieved, then the network can be tested. Otherwise, it has to be trained again, with better or more templates (or input data sets). One pass through all templates is called a training epoch.

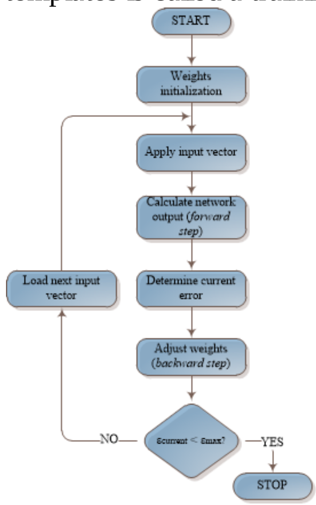


Fig. 5 Backpropagation algorithm

3. ALGORITHM IMPLEMENTATION

The working principle is summarized in Fig. 6.

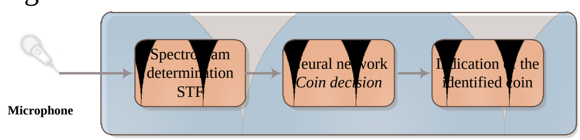


Fig. 6 Software working principle

Coin recognition is realized by identifying the coin's natural frequencies. In order to do this the software must determine the recording's spectrum.

The coin's recording is processed in order to determine its spectrogram by using the short-time Fourier transform with rectangular window. After that, the spectrogram is applied to the input layer of the neural network and the identified coin is indicated.

If the FFT had been applied to the entire recording, then the distinctive spectral components would have been attenuated. To avoid this, STF has been used.

The algorithm implements a multilayer perceptron neural network, with three layers and neurons with unipolar sigmoid transfer functions. Fig. 7 shows a neural network during the training process, in which the recording's spectrogram is applied to its input neurons, and the values of the output neurons are used to adjust the network's weights so that the total error becomes lower that a predefined value.

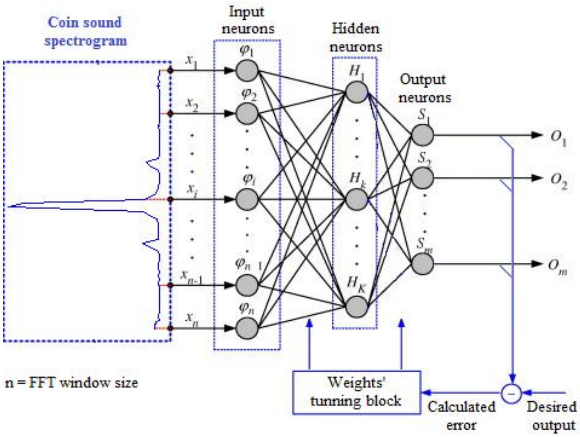
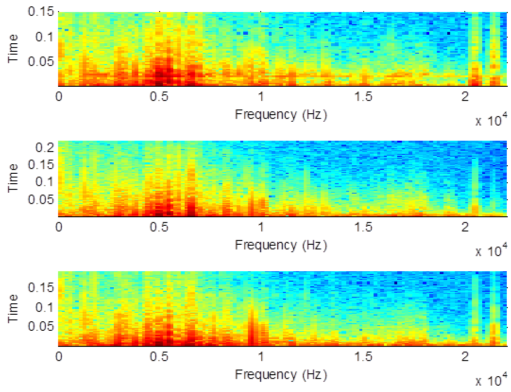


Fig. 7 Neural network during training

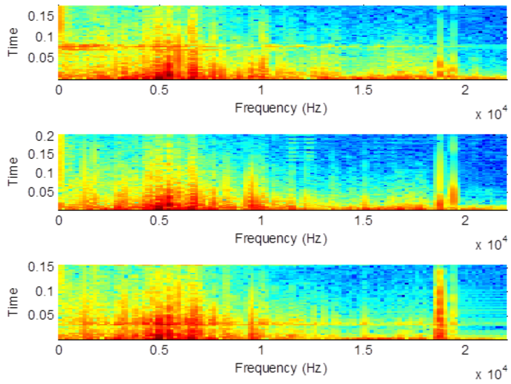
The sound a coin emits when it hits a certain surface depends very much on the material the surface is made of and the position in which the coin contacts it. To avoid these problems a testing stand from wood was built, that contains a solid iron surface, of about 20kg, which will not attenuate the distinctive spectral components included in the sound emitted by a coin, and a slit for inserting the coins and ensuring that they fall on their edge.

4. SIMULATION

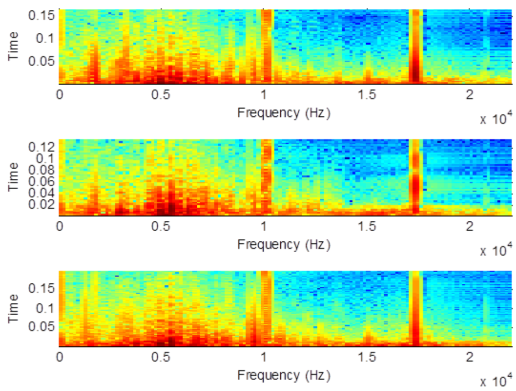
Fig. 8 shows a comparison between the spectrograms of the sounds emitted by a 5 bani coin, a 10 bani coin, respectively a 50 bani coin, obtained in MATLAB using 1024-point short-time Fourier transform. The figures illustrate three successive recordings of each coin, made using a regular PC microphone and an onboard sound card, with a sampling frequency of 44.1kHz.



5 bani coin recording



10 bani coin recording



50 bani coin recording

Fig. 8 Coins recordings' spectrograms

The spectra display the 50 bani coin's natural frequencies at about 10kHz, respectively 18kHz. The spectral components below 10kHz are due to the ambient noise.

Like the spectrograms show, smaller coins have higher natural frequencies. The largest coin, 50 bani, has the lowest distinctive spectral components, while the smallest coin, 5 bani, has the highest components. The 1 bani coin is the smallest one tested and its natural frequencies exceed the 22.05kHz boundary, because the recordings were made using a 44.1kHz sampling frequency.

In order to simulate the neural network, the algorithm was implemented in MATLAB language and a GUI was developed. Fig. 9 shows the GUI flow chart.

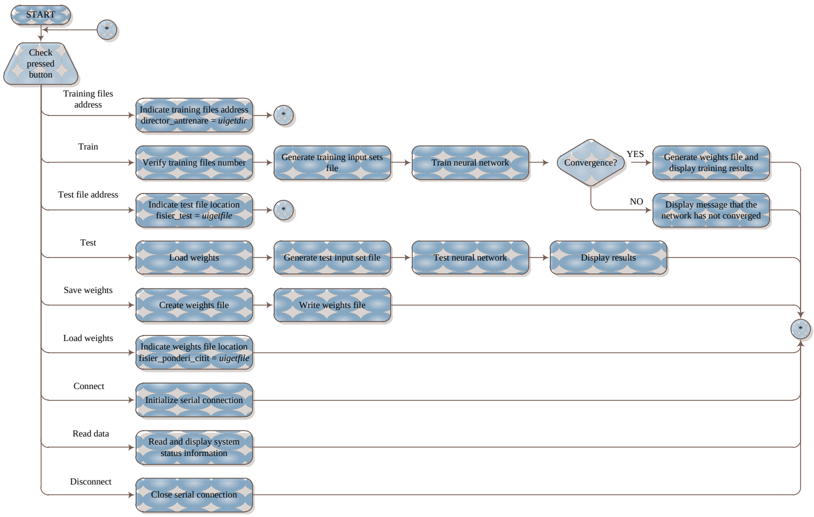


Fig. 9 MATLAB GUI flow chart

Fig. 10 shows the user interface after training the neural network, displaying the process' results.

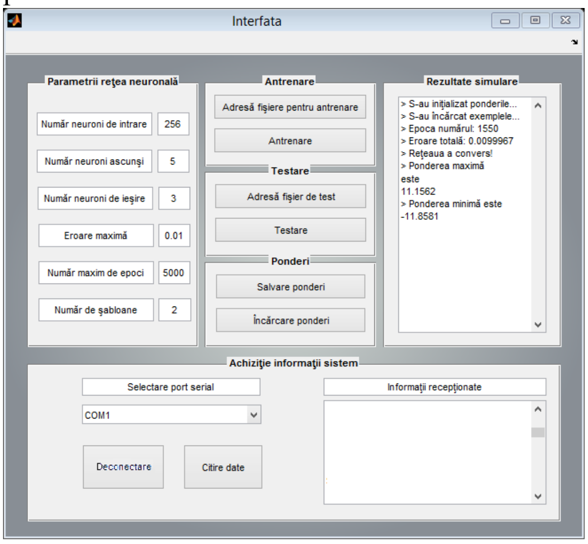


Fig. 10 MATLAB GUI after neural network training

The above figure indicates that convergence was achieved for a 256-5-3 network (256 input neurons, 5 hidden neurons, 3 output neurons), with a total error less than 0.01, using two input data sets for each coin.

After the network parameters are introduced and the recordings directory is configured, the network can be tested. When testing the network, in order to have a good degree of certainty that the coin is identified correctly, a decision threshold of 85% has been defined. In other words, the network has to output a value greater that 0.85 on an output neuron to indicate that a coin has been recognized.

Fig. 11 shows that the network which previously converged has recognized a coin with 100% accuracy. This is because a training input set was applied to the network in the testing process.

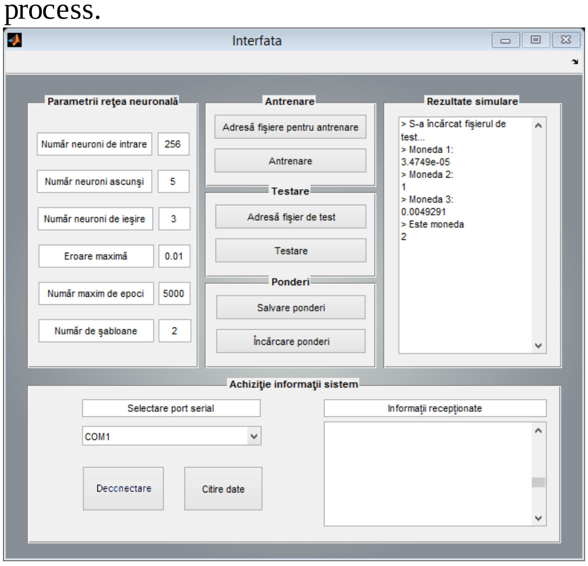


Fig. 11 MATLAB GUI after neural network testing

The GUI also presents a serial communication feature which displays the information acquired from an electronic system which detects coins. The development of this system is a future work goal.

After simulating the network with different parameters, several observations have been made:

- the increase in number of hidden neurons increases the convergence time, but decreases the number of training epochs required for convergence (Table 1);

Table 1 Influence of number of hidden neurons on network convergence time

Number of hidden neurons	Number of training epochs required for convergence	Time required for convergence [s]
1	The network did not converge	-
2	The network did not converge	-
3	1704	~1,5
4	1662	~2
5	1580	~2
10	1499	~3
20	1404	~5
50	1213	~11
100	1089	~17

Fig. 12 shows that the number of training epochs decreases as the number of hidden neurons increases.

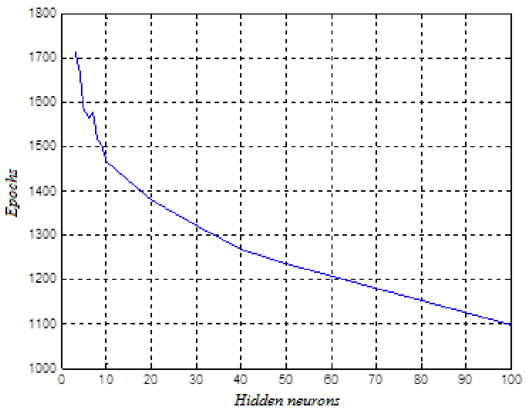


Fig. 12 Number of training epochs depending on the number of hidden neurons

The number of training epochs is dependent on the values with which the weights are initialized, the number of input and output neurons and the total error. Table 1 contains the mean value obtained from five successive trainings of a network with 128 input neurons, 3 output neurons, a total error of 0.01, 5000 total training epochs and two training sets per coin.

- the increase in number of input neurons decreases the number of training epochs required for convergence;

The following test conditions were considered: 10 hidden neurons, 3 output neurons, total error of 0.1, two input sets per coin. The graph displayed in Fig. 13 shows the mean values from 5 successive simulations.

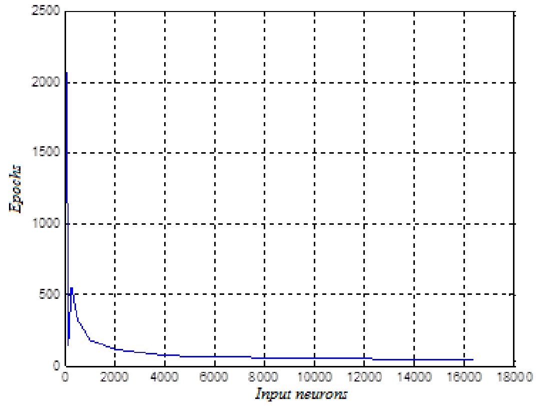


Fig. 13 Number of epochs depending on the number of input neurons

This decrease in number of training epochs is due to the increase of spectrum resolution. In other words, the coins recordings' spectra are easier distinguished if they are defined more precisely (in more points).

- the increase of the total error decreases the number of training epochs required for convergence (Fig. 14).

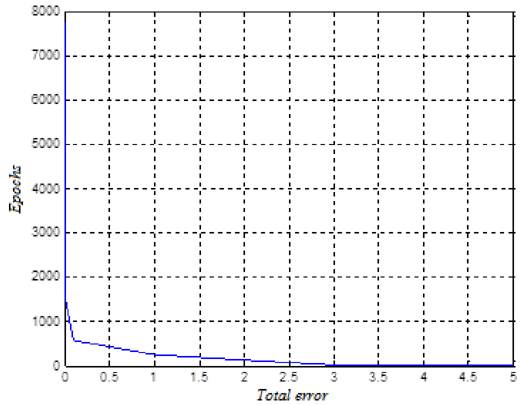


Fig. 14 Number of training epochs depending on the total error

The algorithm converges faster when the total error is smaller, because the differentiation between the output classes is done with lower precision.

Fig. 15 shows the learning curve for a 128-5-3 network, with the total error of 0.01, indicating that convergence is achieved in about 500-600 training epochs.

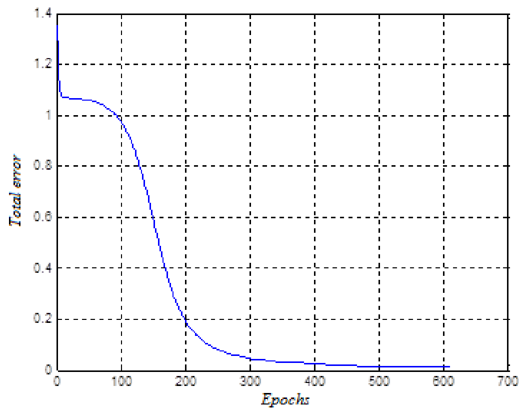


Fig. 15 128-5-3 neural network learning curve

Simulations using the GUI have indicated that the very good results are obtained when training the network with one template per coin. The algorithm successfully identifies all the coins. The 1 ban coin recognition can be explained through the fact that its recording's spectrum is different from the other spectra. It does not contain the natural frequencies, because they exceed the 22.05kHz boundary, making it different from the other spectra. The neural network correctly distinguishes this difference.

5. CONCLUSIONS

The algorithm developed permits the identification of coins by the sound generated when they hit a hard surface. It ensures flexibility, by allowing the user to configure which numerical sequences to identify. In other words, the neural network can be reconfigured to identify other types of coins, not just Romanian coins.

The MATLAB GUI is a useful tool for studying the algorithm by easily configuring different parameters. It can also be used to connect to an electronic system which detects coins in order to read its status information (like the number of coins processed or the number of invalid coins identified) through serial interface.

Training the network with just one input data set for each coin is sufficient for correct identification of coins, and convergence is achieved in 800-1200 training epochs. Training with one template per coin speeds up the training process.

The principle applied in this project can be used for the recognition of other numerical sequences, produced by other objects.

The algorithm can be easily implemented on a DSP.

From the hardware point of view, future work will focus on creating an electronic system which can detect coins.

From the software point of view, future plans include testing different types of neuron transfer functions and of neural networks to evaluate their performances. Also, the system will be tested using different spectral resolutions (by modifying the number of points in which STF is calculated), and using a microphone with better frequency response (broader spectrum sensitivity), which will provide better recordings. Testing different artificial intelligence algorithms, meaning implementation of fuzzy or expert systems, is taken into account.

PERMISSIONS, COPYRIGHT TRANSFER OR ACKNOWLEDGMENT

The manuscripts should be sent to: IONESCU VALERIU, UNIVERSITY OF PITESTI, Târgul din Vale, No.1, 110040, Piteşti, Argeş, ROMANIA, PHONE / FAX: 0248 222949, **E-MAIL: valeriu.ionescu@upit.ro.**

REFERENCES

[1]. STEED, E., "*Coin Identification through Natural Frequency Analysis*", p. 1, available at: http://me363.byu.edu/sites/me363.byu.edu/files/Emerson_Steed_CoinIdentification.pdf.

[2]. ENE, A., "*Rețele neuronale – Teorie și aplicații C*", Tip Naste, Pitești, 2001.

[3]. DREYFUS, G., "*Neural Networks – Methodology and Applications*", p. 28, Springer-Verlag Berlin Heidelberg, 2005.