Identifying Olive (*Olea europaea*) Cultivars Using Artificial Neural Networks

S. Mancuso and F.P. Nicese
Dipartimento di Ortoflorofrutticoltura, University of Florence, via Donizetti 6, 50144 Florence, Italy

**Abstract.** Backpropagation neural networks (BPNNs) were used to distinguish among 10 olive (*Olea europaea* L.) cultivars, originating throughout the Mediterranean basin. Identification was performed on the basis of 17 phyllometric parameters resulting from image analysis. Different BPNN architectures were attempted and best performance was achieved using a $17 \times 20 \times 10$ BPNN. Networks were tested with sets of phyllometric parameters not involved in the training phase. Results enabled identification with certainty all cultivars tested.

Olive (*Olea europaea*) is a species of great economic importance in the entire Mediterranean basin, where 95% of the world production occurs. A large number of cultivars exist, many of which are of ancient origin and genetically heterogeneous, creating a problem of cultivar identification and classification. Present systems used to identify olive cultivars are based on quantitative and/or qualitative parameters (Barranco and Rallo, 1985; Di Prima, 1949). Statistical methods, such as discriminate functions (Damigella, 1960), have been used in the analysis of the above parameters. Recently, interesting results in the area of olive cultivar identification have resulted from isozyme analyses (Ouazzani et al., 1993; Trujillo et al., 1990, 1995), and random amplified polymorphic DNA (RAPD) markers (Bogani et al., 1994; Fabbri et al., 1995). Although these methods are effective, they are resource and labor-intensive, and require a skilled and experienced technical staff to be exploited effectively. Therefore, we assessed the use of artificial neural networks (ANN) as a possible alternative for cultivar distinction and identification in olive.

ANNs are processing devices that are loosely modeled after the neuronal structure of the brain, reflecting the hypothesis that information in a brain resides in the strength of connections between neurons, not in their internal state. An ANN consists of a pool of simple processing units (nodes or neurons), which communicate by sending signals to each other over a large number of weighted connections. Each unit performs a relatively simple job, receive input from neighbors or external sources and use it to compute an output signal that is propagated to other units. The system is inherently parallel in the sense that many units can carry out their computations at the same time. Most neural networks have some sort of learning rule whereby the weights of connections are adjusted on the basis of data. In other words, neural networks learn from examples. This means they must be taught using a set of training data where known solutions are supplied.

The most widely used learning method for neural network is the backpropagation of error. A backpropagation neural network (BPNN) in its basic form is composed of layers of neurons: an input layer, one or more hidden layers, and an output layer. Each layer of neurons receives its input from the previous layer or from the network input. The output of each neuron feeds the next layer or the output of the network. Each layer also has an additional element called a bias unit. Bias nodes simply output a bias signal to the nodes of the current layer (for explanations on the bias units see Rumelhart and McCleland, 1988). This is illustrated in Fig. 1, which shows a three-layer BPNN. Bishop (1995) and Ripley (1996) have written comprehensive reviews on neural networks.

Applications of ANNs in horticulture are just beginning; their use in other areas of science and technology has advanced farther (e.g., voice and handwriting recognition, vibration analysis, diagnostic use in medicine, and elemental particle identification in physics). The most useful qualities of ANNs, such as their skill and speed in recognizing pattern and shapes (Hertz et al., 1991), have barely been exploited in horticulture. ANNs have a structure that makes them particularly useful in recognition of patterns in complex, nonlinear data, such as that derived from many experimental areas of horticulture.

Mancuso et al. 1998, recently conducted a study using ANNs for cultivar identification in grape (*Vitis vinifera* L.). Therefore, we decided to expand on this work by testing the possibility of applying ANNs to distinguish among olive cultivars by using quantitative morphological traits originating from image analysis of leaves.

**Materials and Methods**

**Plants.** Ten olive cultivars (Table 1) were chosen from a...
germplasm collection maintained in a 6-year-old orchard in Pescia, Italy (43°54'N, 10°41'E). Fully expanded leaves located between the seventh and eleventh nodes from the shoot tip were sampled from 10 different plants per cultivar randomly selected in the field, and used for image analysis measurements. Leaf images were obtained at 300 x 300 dpi, 256 gray scale, by using an optical scanner (Epson GT 5000, Torrance, Calif.). Seventeen parameters were determined for each leaf (Table 2) by way of image analysis performed on a P200 computer using the UTHSCSA Image Tool program (developed at the University of Texas Health Science Center, San Antonio, Texas and available from the Internet by anonymous FTP from maxrad6.uthscsa.edu).

**NEURAL NETWORK DETAILS.** A backpropagation neural network program was written, based on a algorithm specifically developed by us. However, a great number of algorithms, that could be used for this analysis, are available. Many of them can be obtained from the Internet free or on commercial basis.

As backpropagation of error is well described in the literature (Rumelhart et al., 1986) we will only provide the final formulas used in this study for correction of weights. The adjustment to the weights most frequently used was a simple gradient descent: each weight adjusts by a small amount proportional to the derivative of the error function (\(dE/dw_i\)) with respect to that weight, and in the opposite direction:

\[
\Delta w_i = -\varepsilon (\delta E/\delta w_i)
\]

\(\Delta w_i\) is the adjustment of the weight \(w_i\) connecting the unit \(i\) of one layer with the unit \(j\) of the next layer and \(\varepsilon\) is the learning rate usually variable from problem to problem in the range 0.001 to 1.0. The present study used a strict gradient descent (Rumelhart and McClelland, 1988) introducing an averaging term:

\[
\Delta w_{ij} (\text{iteration } n) = -\varepsilon (\delta E/\delta w_{ij}) + \mu \Delta w_{ij} (\text{iteration } n-1)
\]

where the coefficient \(\mu\) is referred as the momentum.

The RMS (root mean square) error between the network response (\(X_{P,K}\)) and the training targets (\(T_{P,K}\)) was computed after each iteration according to the equation:

\[
\text{RMS Error} = \sqrt{\frac{\sum_{P,K}(T_{P,K} - X_{P,K})^2}{PTKT}}
\]

where the coefficient \(\mu\) is referred as the momentum.

In total, data from 800 leaves (80 per olive cultivar; 40 in the learning phase and 40 in the recognition phase) were used. As ascertained in preliminary tests, 40 represents the minimum significant number of leaves for each phase. On the other hand, any increase in the number of the samples does not improve the final result. Both learning and recognition phase of the BPNN were achieved on a P200 computer. The learning phase in all the BPNNs tested was protracted until the RMS was <0.07 and the difference between the RMS in two consecutive epochs was <0.0001. The ANNs were tested with sets of phyllometric parameters in inputs for which the output was known, so that the predicted and actual outputs could be compared. These data had not been used previously to train the network.

**Results and Discussion**

**DETERMINING NETWORK TOPOLOGY.** Preliminary, different BPNN architectures were attempted, to verify the best network topology. Three main questions were posed: 1) Should the

Table 1. List of the olive cultivars and number of leaf samples used in this research.

| No. | Cultivar       | No. of samples |
|-----|---------------|----------------|
| 1   | Ascolana tenera | 80             |
| 2   | Frantoio      | 80             |
| 3   | Leccino       | 80             |
| 4   | Leccio del Corno | 80         |
| 5   | Maurino       | 80             |
| 6   | Minerva       | 80             |
| 7   | Moraiolo      | 80             |
| 8   | Pendolino     | 80             |
| 9   | Picholine     | 80             |
| 10  | Vescovo       | 80             |
| Total |                | 800            |

Table 2. Leaf parameters considered by neural networks.

| No. | Parameter      | Definition                                           |
|-----|----------------|------------------------------------------------------|
| 1   | Area           | Area of the leaf                                    |
| 2   | Perimeter      | Perimeter of the leaf                               |
| 3   | Major axis length | Length of the longest line that can be drawn through the leaf |
| 4   | Minor axis length | Length of the longest line that can be drawn through the leaf perpendicular to the major axis |
| 5   | Roundness      | Computed as: \(4\pi \times \text{area} \times \text{perimeter}^{-2}\) |
| 6   | Elongation     | Ratio of the length of the major axis to the length of the minor axis |
| 7   | Feret diameter | Diameter of a circle having the same area of the leaf |
| 8   | Compactness    | Computed as: \(\sqrt{4 \times \text{area} \times \pi^{-1} \times \text{major axis length}^{-1}}\) |
| 9   | Centroid       | Center of mass of the leaf. It is computed as the average of the x and y coordinates of all the pixels in the leaf below, and to both sides of the point. It is computed as a weighted average of the x and y coordinates and the pixel brightness for all pixels in the image of the leaf |
| 10  | Gray centroid  | Also called the brightness-weighted center of mass, this is the point in the object having equal brightness levels above, below, and to both sides of the point. It is computed as a weighted average of the x and y coordinates and the pixel brightness for all pixels in the image of the leaf |
| 11  | Integrated density | Computed as the product of the mean grey level and the number of pixels in the image of the leaf |
| 12  | Min gray level | Minimum gray level of the leaf                       |
| 13  | Mean gray level | Mean gray level of the leaf                          |
| 14  | Median gray level | Median gray level of the leaf                        |
| 15  | Mode gray level | Mode gray level of the leaf                          |
| 16  | Max gray level  | Maximum gray level of the leaf                       |
| 17  | SD gray level   | Standard deviation of the gray level of the leaf     |
network have one or nine units in the output layer? 2) Which activation function should be used? and 3) How many units should be in the hidden layer?

The first question was solved by comparing results on classification of the networks with one output (1, 2, and 3), with results of the networks with 10 outputs in Table 3. BPNNs with 10 outputs performed better than those with one.

To address the second question, two different activation functions were tried: sigmoid and gaussian. A node’s activation function controls the output signal strength for the unit (except for the input layer, which uses the inputs themselves). These functions set the output signal strength between 0 and 1. The sigmoid function, represented by the mathematical relationship \(1/(1+e^{-x})\), is the most widely used function for backpropagation neural networks. The sigmoid function acts like an output gate that can be opened (1) or closed (0). Since the function is continuous, it is also possible for the gate to be partially opened (i.e., somewhere between 0 and 1). Models incorporating sigmoid transfer functions usually exhibit better generalization in the learning process and often yield more accurate models, but they can also require longer training times (Ripley, 1996). The gaussian transfer function can greatly alter the dynamics of a neural network model. Like the sigmoid function, the output response is normalized between 0 and 1, but the gaussian transfer function is more likely to produce an in-between state. Gaussian based networks tend to learn more quickly than sigmoid counterparts (compare for example in Table 3, BPNN 4 and 5 or BPNN 6 and 7), but also tend to produce networks that are prone to memorization (the network tend to memorize input and output sets rather than learn relationships between them). Table 3 shows the error on classification obtained using sigmoid or gaussian activation functions: the best classification was obtained with the sigmoid function (BPNN number 6); to achieve the same error using the gaussian function, more hidden layers were needed (BPNN number 9), with an increased time for the learning process.

Finally, construction of the hidden processing structure of the network is arbitrary. Many factors play a part in determining what the optimal configuration should be. These factors include the quantity of training patterns, the number of input and output nodes and the relationships between the input and output data (Zurada and Jacek, 1992). Having many hidden layers and processing units will result in a model that performs poorly. When a network’s hidden processing structure is too large and complex for the model being developed, the network may tend to manifest the memorization problem described above. Such a network may train well but test poorly when presented with inputs outside the training set. Results in Table 3 show that the best arrangement was reached using 20 hidden units (BPNN 6). A larger network (BPNN 8) with a double hidden layer, was less successful in classification, demonstrating quite well that larger networks do not necessarily provide better performance.

**IDENTIFYING OLIVE CULTIVARS.** During the recognition phase, BPNN number 6 (Table 3), was able to identify all the cultivars used to test it (Fig. 2). Forty leaves for each cultivar were tested, and they always gave the greatest output in the expected class. In an ideal case, only one output class would have the averaged (for the 40 leaves) output signals equal to 1, while the other output classes would have an output of zero. In practice, however, the actual output does so only sporadically, with those classes that are expected to be zero being only close to it, and the correct class ranging from 0.5 to 1.0. So, for example, testing the ‘Ascolana’ cultivar in Fig. 2, we had the greatest output in the correct class ‘Ascolana’, with an averaged value of 0.85, whereas the other classes gave averaged values ranging from 0.000129 to 0.041.

As seen in Fig. 2, the BPNN outputs of ‘Maurino’, ‘Picholine’ and ‘Pendolino’, as well as ‘Minerva’ and ‘Moraiolo’, produced results showing these cultivars to be loosely related. ‘Minerva’, is a patented clone of ‘Leccio’; ‘Leccio del corno’ (as the name suggests) is commonly considered a ‘Leccino’ clone as well. However, the data obtained in this study show no apparent relationship among these olive accessions. All the other accessions produced results showing them to be clearly differentiated. Some, such as ‘Ascolana’ or ‘Vescovo’, showed a BPNN output so defined as to resemble an ideal case output.

Results herein demonstrate that backpropagation neural networks can be used effectively to differentiate olive cultivars through phyllometric parameters. Though one of the acknowledged advantages of BPNN is the capability to overcome the need for a statistically representative sample of a population (Dawn, 1994), we can’t exclude that for a real practicable system the environmentally changed morphology must be taken into consideration. As a matter of fact, further research should be conducted to assess the stability of the BPNN with samples coming from very different environments. Moreover, additional research is needed to test the suitability of BPNN to distinguish among clones within the same cultivar, after the promising results obtained with distinction of the ‘Minerva’ clone. On the other hand, it should be emphasized that no sophisticated instrumentation is required. Analyses can be performed with a medium-performance computer easily available in any university department.
Fig. 2. Outputs of the neural network recognition phase. Each frame shows the backpropagation neural network (BPNN) output for the input represented by the phyllometric parameters of 40 leaves. Each line represents a data point from an individual leaf. The name of the unknown cultivar is given by the cultivar number (as in Table 1) on the abscissa, which present the highest (closest to 1) output value. For sake of clarity, dashed lines showing the averaged output data, are reported. Vertical bars represent SD.
Literature Cited

Barranco, D. and L. Rallo. 1985. Las variedades de olivo cultivadas en España. Olivae 9:16–22.

Bishop, C.M. 1995. Neural network for pattern recognition. Oxford Univ. Press, Oxford, U.K.

Bogani P., D. Cavalieri, R. Petruccelli, L. Polsinelli, and G. Roselli. 1994. Identification of olive tree cultivars by using random amplified polymorphic DNA. Acta Hort. 356:98–101.

Damigella, P. 1960. Variabilità dei caratteri biometrici dell’olivo e impiego delle “funzioni discriminanti.” La ricerca Scientifica 30:18–26.

Dawn, T. 1994. Neural computing makes its mark in science. Scientific Computing 3:25–30.

Di Prima, S. 1949. Primo contributo allo studio biometrico delle varietà di olivo in provincia di Bari. Ann. Della Sper. Agr. 8:457–491.

Fabbri A., J.I. Hormaza, and V.S. Polito. 1995. Randomly amplified polymorphic DNA analysis of olive (Olea europaea L.) cultivars. J. Amer. Soc. Hort. Sci. 120:538–542.

Hertz, J., A. Krogh, and R. Palmer. 1991. Introduction to the theory of neural computation. Addison-Wesley, Redwood City, Calif.

Mancuso S., P.L. Pisani, R. Bandinelli, and E. Rinaldelli. 1998. Application of an artificial neural network (ANN) for the identification of grapevine genotypes. Vitis 37:27–32.

Ouazzani, N., R. Lumaret, P. Villemur, and F. Di Giusto. 1993. Leaf allozyme variation in cultivated and wild olive trees (Olea europaea L.). J. Hered. 84:34–42.

Ripley, B.D. 1996. Pattern recognition and neural networks. Cambridge Univ. Press, Cambridge, U.K.

Rumelhart, D.E, G.E. Hinton, and R.J. Williams. 1986. Learning representations by back-propagating errors. Nature 323:533–536.

Rumelhart, D.E. and J.L. McClelland. 1988. Exploration in parallel distributed processing. MIT Press, Cambridge, Mass.

Trujillo I., L. Rallo, and P. Arus. 1995. Identifying olive cultivars by isozyme analysis. J. Amer. Soc. Hort. Sci. 120:318–324.

Trujillo I., L. Rallo, E.A. Carbonell, and M.J. Asins. 1990. Isoenzymatic variability of olive cultivars according to their origin. Acta Hort. 286:137–140.

Zurada F. and M. Jacek. 1992. Introduction to artificial neural systems. PWS Publishing, Pacific Grove, Calif.