Linear and Non-linear Pattern Recognition Models for Classification of Fruit from Visible-Near Infrared Spectra

Jaesoo Kim, Alistair Mowat, Philip Poole and Nikola Kasabov

Address for correspondences:

Jaesoo Kim

School of Natural and Rural Systems Management

The University of Queensland

Gatton College, QLD 4345

Australia

Fax : +61-7-5460-1324

Linear and Non-linear Pattern Recognition Models for Classification of Fruit from Visible-Near Infrared Spectra

Jaesoo Kim[1], Alistair Mowat[2], Philip Poole[3], and Nikola Kasabov[1]

[1]University of Otago, Information Science Department, P.O. Box 56, Dunedin, New Zealand

[2]Horticulture and Food Research Institute Ltd, Ruakura Research Centre, PB 3123, Hamilton, New Zealand

[3]Poole Scientific Ltd, 48 Southsea Cresent, Hamilton, New Zealand

Abstract— *Environment and genotype affect the composition, quality, storability and sensory properties of plant based-products. Visible-near infrared (NIR) spectral measurements is used increasingly to monitor fruit properties such as maturity, sensory properties and storability non-destructively both prior to harvest and during storage. To explore this problem, at harvest and after storage, visible-NIR spectra containing 1024 individual data points were measured on kiwifruit berries sourced from six pre-harvest fruit management treatments. These raw spectra were processed by principal component analysis, or by Fourier, Hartley, Haar, Hurst, range renormalisation or polar coordinate transforms in order to extract a smaller set of features selected independently of treatment. In order to reduce their dimensionality further, the extracted features were processed by canonical variate (cv) analysis. The ability of various connectionist and linear discrimination pattern recognition models to predict the treatment source of unknown fruit on the basis of these features were evaluated. Thus far, this work* *has established that the performance of the non-linear model was shown to be significantly better in comparison to the linear model. From these results, it has also been shown that both the feature extraction and selection techniques have a marked effect on the ability to classify fruit by treatment source and storage date. In general, the best classifications were based on features extracted using the FFT method, but the best performance in any single classification was given by the Haar transform in conjunction with the scaled conjugated gradient learning method.*

Keywords—VNIR Spectra, Kiwifruit, Linear discrimination, Artificial neural networks, Feature extraction, Pattern recognition and Classification, Canonical variate analysis.

1 INTRODUCTION

Increasingly, visible-near infrared (visible-NIR) spectrophotometry is being used in horticultural research to characterise fruit quality attributes such as water contents and sugar content of the expressed juice (soluble solids) [**?**]. For fresh fruit, an important aspect of this technique has been the ability to non-destructively assess internal quality attributes such as soluble solids concentration, dry matter content, acidity and firmness [**?**]. The technique is rapid, allowing fruit to be graded on commercial packing lines [**?**]. More recently, visible-NIR spectroscopy has been used, on the vine, at harvest or during storage, to distinguish differences between fruit grown under different conditions or to predict storability [**?**].

For the non-destructive prediction of fruit compositional attributes, such as water and sugar content, multivariate linear regression (MLR) or partial least squares (PLS) analysis are used to relate specific spectral regions to changes in the concentration of a known attribute [**?**, **?**]. In kiwifruit (*Actinidia deliciosa* (A. Chev) C.F. Liang *et* A.R.Ferguson var. *deliciosa*), this approach was used to estimate soluble solids content and dry matter in ripened berries [?]. Likewise, similar techniques have been used to predict post-ripening sensory properties in mangoes [?], which are also harvested in an unripe state. In kiwifruit, although dry matter and soluble solids content can be measured by visible-NIR techniques, other properties such as texture [?], flesh coloration [?] and aroma [?] are also important in determining sensory properties.

Alternatively, spectral data of a fruit can be treated as a signature, allowing fruit to be grouped on the basis of their spectral similarities. In the processed food industries, such an approach is used in product control, for example, to detect adulteration of food products such as fruit juices [?], jams [?], coffee [?], and grains [?]. Recently, based on this approach, we have used visible-NIR spectra to differentiate persimmon (*Diospyros kaki L.*) treated with growth regulators during late fruit development and ripening [?] and kiwifruit berries grown under different development conditions.

One of the main problems encountered in product identification and characterisation from such spectra has been the large volume of primary data (typically between 250 and 2000 points per spectrum, depending on the instrumentation used). In the previous work on kiwifruit, the original visible-NIR absorption spectra were reduced to a series of principal components (PC) which were then analysed by canonical variate (cv) and linear discriminant (LD) techniques. It was shown that individual fruit could be distinguished by treatment from the weightings of these components. Furthermore, a preliminary study showed that connectionist-based methods could provide superior classification for these systems [?].

Pattern recognition systems appropriate for fruit spectra are likely to consist of three stages. Firstly, the dimensionality of the raw data is reduced by a feature extraction process to the extent possible while retaining sufficient information to enable the resultant features to be used for classification. For this purpose, principal component analysis (PCA) has been the main technique used [?, ?], but other approaches, notably the Fast Fourier Transform (FFT) [?] and related Fast Hartley Transformation (FHT) [?, ?], have been proposed. The rules for the extraction are normally derived from a randomly selected subset of the whole sample. Unlike the FFT and FHT, the results of the PC analysis are strongly dependent on the variation that exists in the *training* subset.

Wavelet [?] and polar coordinate transforms (PCT) [?] have also been considered as primary feature extraction approaches. The Haar transform (HT) [?], a simple wavelet transformation, by forming the normalised average of adjacent terms, halves the volume of data for each transformation cycle. For polar coordinate transforms, the spectra are represented in a 2D coordinate system, where centres, determined variously from the individual points, the perimeter or the bounding surface can be used to can be used to reduce the spectra to $2 \sim 6$ points [?]. A further technique, that has not previously been applied to spectral analysis, is the rescaled range (RR) approach used in Hurst exponent analysis [?].

The second stage of the pattern recognition process is feature selection. A wide variety of techniques including canonical variate analysis, genetic algorithms (GA) classification and regression trees (CART), and intermediate partial least squares (IPLS) search techniques, have been used to select features by which features suitable for discriminating treatments can be derived [?, ?, ?]. In some systems, these features may be extracted directly from the primary data, but more commonly they are derived from the features extracted above. Whilst the selection of these features may result in a further reduction in dimensionality, a more important aspect is the elimination of wavelength regions and features that are less important to the classification, thus reducing sources of noise, thus resulting in more robust calibration equations [?].

The final stage is pattern recognition, based on extracted or selected features, are used to predict specific fruit quality attributes, such as dry matter, or classify unknown fruit into known groupings, such as high or low storability. In some situations, the extracted features may be satisfactory without further processing, and any further processing is necessarily compared with that option. Traditionally, MLR, PLS, principal component regression (PCR), and LDA have been the main pattern recognition techniques applied to visible-NIR spectral data.

More recently, connectionist-based methods, such as back-propagation (BP), multi-layer perceptrons (MLP), self-organising maps (SOM), GA's and neuro-fuzzy models, and machine learning methods, such as CART, have been successfully applied to visible-NIR pattern recognition problems [?, ?, ?, ?, ?, ?, ?, ?]. Here, research efforts have focused on determining whether specific connectionist approaches could improve upon the traditional calibration methods, particularly in systems where there is expected to be substantial non-linearity [?], and whether they offer any computational advantages.

Generally, connectionist methods have been used in quantitative problems but have recently been applied to qualitative analyses [?, ?, ?]. In either case, multi-layer perceptrons (MLP), usually trained within the generalised delta rule and the back-propagation (BP) algorithm, is the basic architecture [?]. Because the BP algorithm is equivalent to the steepest descent algorithm, these algorithms tend to converge slowly for practical problems.

In this paper we discuss and evaluate several high performance algorithms which can converge faster than the standard BP algorithms. Faster convergence algorithms used in the field of nonlinear optimisation, such as the resilient back-propagation (Rprop) method [?], scaled conjugate gradient (SCG) [?], adaptive learning rate back-propagation (ABP) [?], and quick-propagation method (Qprop) [?] may accelerate the convergence of the BP algorithm.

These faster algorithms fall into two main categories. The first category uses heuristic techniques, which were developed from an analysis of the performance of the standard gradient descent method. One heuristic modification is the momentum technique, which has been successfully used in many pattern recognition problems. However, this paper will discuss another two heuristic techniques: ABP and Rprop.

The second category of fast algorithms uses standard numerical optimisation techniques, i.e. second order optimisation methods, which make use of second derivatives of the error in weight space. In this paper a variation on conjugate gradient (CG) descent, scaled conjugate gradient (SCG), that takes some account of the non-quadratic nature of the error surface in weight space has been considered. Unsupervised pattern recognition methods are an alternative method for qualitative analysis [?]. Supervised methods will, however, be described and evaluated in this work.

The aim of the current work is to evaluate a range of pattern recognition techniques for their use in spectral data processing, and the particular problems of fruit classification. In this regard, this paper compares the performance of a linear pattern recognition technique, LDA, with non-linear techniques based on MLPs with variations on BP learning such as ABP, Rprop and SCG to classify kiwifruit grown under different conditions using a range of features extracted from visible-NIR spectra.

2 Neural Network Classifiers

The supervised learning algorithm is given the training data set consisting of N training data pairs. These training pairs (x_i, y_j) are often called examples, where x_i is an m-dimensional pattern vector, whose components are called features, and

y_j is a known class. The mapping function f, $y = \mathbf{f}(x)$, is obviously not known. The training set represents information about some domain with the frequency used assumption that the features represent only properties of the examples but not the relationships between the examples. The supervised learning algorithm then searches the space of possible hypotheses that best estimates the unknown function f: $x \to y$. This is typified by the use of the multilayer perceptrons (MLP), which is also called multilayer feedforward networks.

The MLP has probably been the most widely used of neural network architectures and is a feedforward network with one or more layers of nodes or neurons between the input and output nodes. The layers between the input and the output are called hidden, intermediate or middle layers and have no connections to the external world. It can be shown that two hidden layers are sufficient to perform any classification task although real problems are often much simpler and can be solved using only one hidden layer. An illustration of a multilayer feedforward net with two output variables and one hidden layer is given in Fig **??**.

Each neuron in connectionist models is a simple computational device which calculates the weighted sum of its input signals to get the net input. From this net input it calculates the output signal using a nonlinear function which is sometimes called *squashing function* (also called transfer function or activation function). In general the output function can be of any nonlinear type, although the training method used in most of the networks, performs best with the sigmoid function, by which the weight changes can be calculated in a very easy way. The sigmoid signal processing in a node or neuron is illustrated in Fig. **??**.

The knowledge of a network is represented by its topology and the weights of the neuronic inputs. Given a certain topology the network can change and adjust its knowledge by adjusting its weights according to the presented samples of data. In other words the network creates a nonlinear mapping of the input

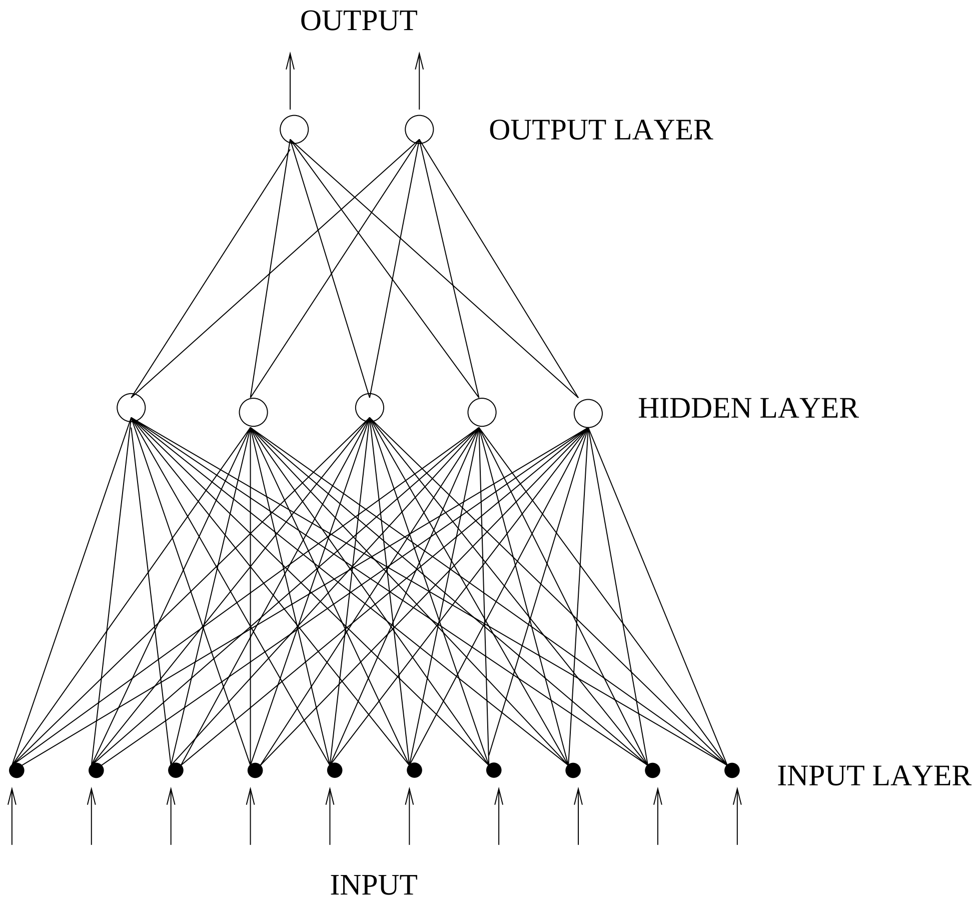

Figure 1: A typical multilayer feedforward network.

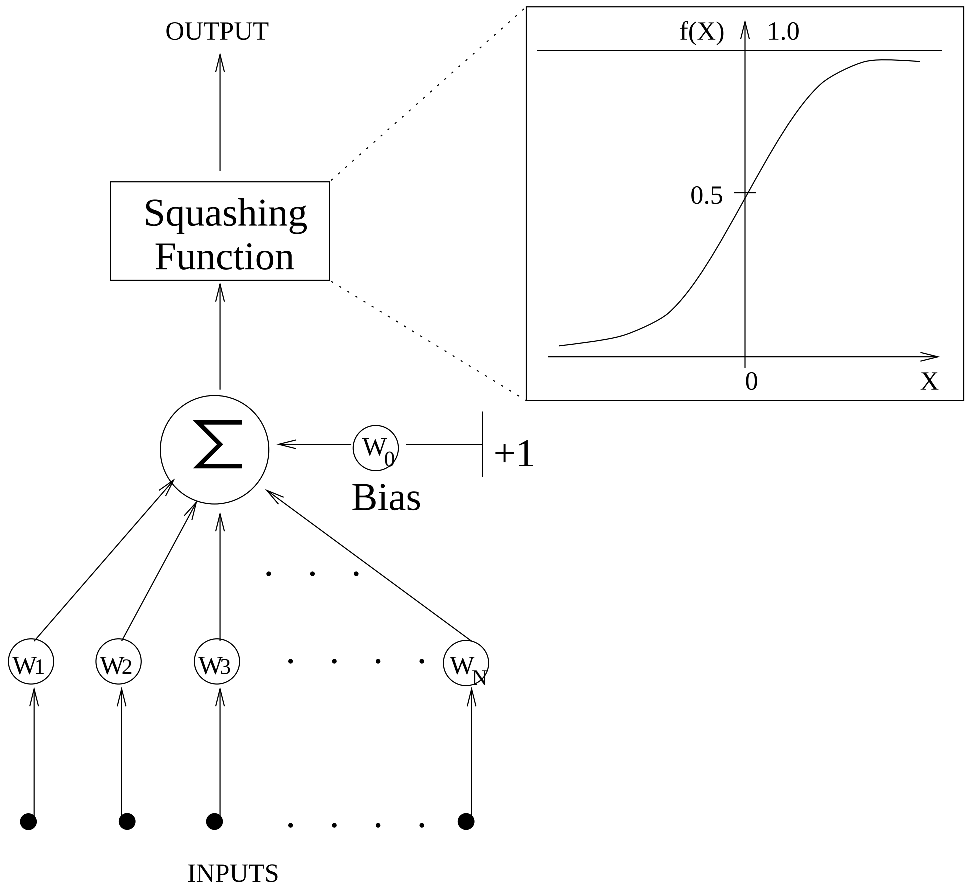

Figure 2: A typical neuron with a sigmoidal function: $f(x) = 1/(1 + \epsilon^{-(x+T)})$.

space to the output space by adjusting its weights in a proper way. A commonly used process involves the repeated exposure of a network to inputs and associated outputs. By allowing the network to examine its mathematical error in the light of a true outputs, the system is able to arrive at some form of optimal weighting arrangements. When learning is complete, stimulation of the network with a set of input values enables it to produce outputs consistent with what it has experienced during learning.

The training algorithm normally used for the MLP is called back-propagation (BP) of errors [**?**, **?**]. The supervised BP model is the most commonly implemented paradigm today because it is the best generalised model. The BP algorithm is a generalisation of the LMS algorithm (or Delta rule) and provides a theoretically sound method for training multilayer feedforward networks. BP is the canonical feed-forward network where an error signal is fed back through the network, altering weights as it goes and is a numerically intensive technique, and there are many different ways to perform BP to teach the artificial neural networks (ANNs) how to respond.

There are numerous algorithms available to be used for training the ANNs, such that it remains necessary to select appropriate learning rule in order to achieve an acceptable solution. The starting point for the derivation of a training algorithm is the definition of an error (cost) function which will be minimised by adapting the network weights. Almost all gradient descent training algorithms minimise the total summed squared error (SSE) function and apply the well-known BP algorithm. Starting with the output layer, BP repeatedly applies the chain rule to the SSE in order to compute error values in the hidden units. By means of these error values the connection weights gradient can be computed by the so called *generalised delta rule*. Unfortunately it can be very slow convergence for practical applications.

The simple gradient descent algorithm proposed in has been modified by various heuristics in order to speed up the convergence and to improve the performance of the trained networks. BP and its variants learn by doing gradient descent using the partial first derivative of the error with respect to each weight. Since we do not know how the error surface is constructed we have to take small steps to keep from overshooting as we move down the error surface. If we knew something about curvature of the error function, we could use this information to decide how large a step to take and in what direction. We can get this information by using the second derivative of the error with respect to the weight.

Over the last few years many improvement strategies have been developed to speed up BP learning. Today's most successful BP variants are adaptive learning rate BP (ABP) [**?**], resilient back-propagation (Rprop) [**?**], and scaled conjugate gradient (SCG) [**?**]. Although the used heuristics can not be theoretically justificated, they outperform even the most sophisticated algorithms from the field of nonlinear optimisation. The following briefly describes an overview of three different speed up techniques, which are known most efficient variants of the BP learning algorithms. The standard back-propagation algorithm was evaluated and described by several researchers [**?**, **?**, **?**, **?**, **?**].

2.1 ABP

It is important to see BP as an algorithm for computing $\frac{\partial E}{\partial w}$. It is not an optimisation procedure in itself, although it is a simple step to implement a gradient descent optimisation procedure from the derivatives furnished by BP. However, other optimisation procedures can use the gradients computed by BP. These optimisation procedures can work faster than steepest gradient descent.

The simple approach to finding suitable weights is to follow the gradient of the error surface in weight space to minimise some error criteria E. Let the set of weights at iteration n be denoted by $w(n)$. Then the update rule for gradient descent is given as follows. In the batched mode variant the descent is based on the gradient ∇E for the total training set:

$$\Delta w(n) = -\eta \frac{\partial E}{\partial w}|_n + \alpha \Delta w(n-1) \tag{1}$$

where $\frac{\partial E}{\partial w}|_n$ is the vector of gradients of the total error E with respect to each weight w evaluated at iteration n, $\Delta w(n-1)$ refers to the most recent weight change, and η (< 1) and α (< 1) are two non-negative constant parameters called learning rate or step size and momentum, respectively. The value of the learning rate, or step size, is crucial for the success of the algorithm.

A small step size leads to slow learning and the possibility of getting trapped in local minima of the error surface. A large step size can overshoot the minimum. The weights may then be set to a point in weight space on a high plateau of the error surface. A plateau in the error surface will exist where the nodes have saturated activations. Here, the gradient is small and the small gradient then means the large step size is irrelevant and the network learns slowly again, but this time with a high error. Thus both small and large step sizes are undesirable. The best step size depends on the error surface, itself a function of the architecture of the network(e.g. the number of nodes and the connections between them) and the training data. Adding a momentum term to gradient descent alters the search direction by adding some of the previous search directions to the current gradient. In practice, the momentum can speed up training in very flat regions of the error surface and suppresses weight oscillation in step valleys or ravines. Unfortunately it is necessary propagate the whole training set through the network for calculating ∇E. This can slow down training for bigger training set. Therefore the update is based just on the gradient for the actual training pattern ∇E_p:

$$\Delta w(n) = -\eta \frac{\partial E_p}{\partial w}|_n + \alpha \Delta w(n-1) \tag{2}$$

As can be seen these methods offer the benefit of simplicity, but their performance depends sensitively on the parameters η and α. A good choice of η and α is very essential for training success and speed. Adjusting these parameters by hand can be very difficult and might take a long time for more complicated tasks. Unfortunately, BP learning algorithms may be also limited by their poor scaling behaviour. As the size of the network increases, the network becomes more computationally intensive, and so the time required to train the network grows exponentially and the learning process becomes unacceptably slow. What is required is a more efficient procedure. This circumstance has given rise to a plethora of heuristics for adaptive variable step size algorithms [?, ?, ?, ?, ?, ?, ?].

Most of gradient descent techniques have their roots in the well-explored domain of optimisation theory. These techniques can roughly be divided into two categories. Algorithms that use global knowledge of the state of the entire network, such as the direction of the overall weight-update vector, are referred to as *global* techniques [?]. There are many examples where adaptive learning algorithms make use of global knowledge: steepest descent used in the standard BP algorithm and the conjugate gradient methods [?, ?, ?, ?, ?].

By contrast, local adaptation strategies such as Qprop [?] and Rprop [?] are based on weight-specific information only, such as the temporal behaviour of the partial derivative of this weight. The local approach is more closely related to the neural network concept of distributed processing in which computations can be made in parallel. Furthermore, it appears that for many applications local strategies work far better than global techniques, although they use less information and are often much easier and faster to compute [?].

With standard gradient descent, the learning rate (η) is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate (or step size). It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface.

The performance of the gradient descent algorithm can be improved if we allow the learning rate to change during the training process. An adaptive learning rate will attempt to keep the learning step size as large as possible while keeping learning stable. The learning rate is made responsive to the complexity of the local error surface.

An adaptive learning rate requires some changes in the training procedure. First, the initial network output and error are calculated. At each epoch new weights are calculated using the current learning rate. New outputs and errors are then calculated as follows. The learning rate (η) is varied according to whether of not an iteration decreases the performance index. If an update results in reduced total error, η is multiplied by a increasing factor (> 1) for the next iteration. If a step produces a network with a total error more than a few percent above the previous value, all changes to the weights are rejected, η is multiplied by a decreasing factor (< 1), momentum (α) is set equal to zero, and the step is repeated. When a successful step is then taken, α is reset to its original value.

The rational behind this maneuver is that as long as the topography of the terrain is relatively uniform and the descent is in a relatively smooth line, the memory implicit in α will aid convergence. The recommended ratio to increase learning rate and to decrease learning rate is 1.05 and 0.7, respectively.

2.2 Rprop

MLP networks typically use squashing functions in the hidden layers, since they compress an infinite input range into a finite output range. Sigmoid functions are characterised by the fact that their slope must approach zero as the input gets large. This causes a problem when using steepest or gradient descent to train a MLP network, since the gradient can have a very small magnitude, and therefore cause small changes in the weights and biases, even though the weights and biases are far from their optimal values.

On the other hand, resilient back-propagation (Rprop) [?] uses the local topology of the error surface to make a more appropriate weight change, which evolves during the learning process according to its local view of the error function. Rprop is very powerful and efficient because the size of the weight step taken is no longer influenced by the size of the partial derivative. It is uniquely determined by the sequence of the signs of the derivatives, which provides a reliable insight about the topology of the local error function. Only the sign of the derivative is used to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate update value. These can be eliminate the harmful effects of the magnitudes of the partial derivatives.

Rprop [?] is the fastest training algorithm and a local adaptive learning scheme [?]. Rprop uses different step sizes for each weight. However, it only uses the sign of the local gradient, $\partial e/\partial w$, when updating the weight, not its magnitude. The size of the derivative is taken to indicate the direction of the weight update and the sizes of the weight update are adapted with respect to the sign of the actual and the last derivative. The step sizes are bounded by upper and lower limits in order to avoid oscillation and arithmetic underflow of floating point values.

The following computational scheme of Rprop can be used:

1. Choose some small initial value for every update step size $\eta(0)$.

2. The vector of step sizes η and *Delta* is defined as follows:

$$\eta(i) = \begin{cases} \eta(i-1)\Delta^{+} & : \left.\frac{\partial e}{\partial w}\right|_{i} \left.\frac{\partial e}{\partial w}\right|_{i-1} > 0, \\ \eta(i-1)\Delta^{-} & : \left.\frac{\partial e}{\partial w}\right|_{i} \left.\frac{\partial e}{\partial w}\right|_{i-1} < 0, \\ \Delta_{max} & : \eta(i) \geq \Delta_{max}, \\ \Delta_{min} & : \eta(i) \leq \Delta_{min}, \end{cases} \tag{3}$$

   where Δ_{max} and Δ_{min} limit the size of the step above and below.

3. The weights were updated as follows:

$$\Delta w(i) = \begin{cases} \pm(\left.\frac{\partial e}{\partial w}\right|_{i})\eta(i) & : \left.\frac{\partial e}{\partial w}\right|_{i} \left.\frac{\partial e}{\partial w}\right|_{i-1} \geq 0, \\ 0 & : otherwise, \end{cases} \tag{4}$$

   such that $w(i+1) = w(i) + \Delta w(i)$. A further detail concerns the stored value of the previous gradient

$$\left.\frac{\partial e}{\partial w}\right|_{i-1}.$$

   If, for a particular weight,

$$\left.\frac{\partial e}{\partial w}\right|_{i} \left.\frac{\partial e}{\partial w}\right|_{i-1} < 0,$$

   the value of the stored gradient for that weight would be set to 0 for the next time step.

Recommended values for the parameters are: $\Delta_{max} = 50.0$; $\Delta_{min} = 0.000001$; $\Delta^{+} = 1.2$; and $\Delta^{-} = 0.5$.

2.3 SCG

The basic BP algorithm adjusts the weights in the steepest descent direction (negative of the gradient). This is the direction in which the performance function is decreasing most rapidly. It turns out that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions.

Mòller [?] has introduced a variation on conjugate gradient (CG) descent that takes some account of the non-quadratic nature of the error surface in weight space: the line search part of CG minimisation can be tricky and there exists a variant, *Scaled Conjugate Gradient* (SCG) [?], that avoids the line search by estimating the minimisation step η_t through

$$\eta_i = \frac{-\mathbf{d}_i^T \nabla E_i}{\mathbf{d}_i^T \; H \; \mathbf{d}_i}. \tag{5}$$

This SCG method is usually faster than normal CG since it avoids the line search. Eq. (**??**) has been avoided in the past due to the effort of finding the Hessian H. The Hessian may be approximated by a method of finite differences, or the product $H\mathbf{d}$ may be approximated by taking a difference of gradients [?]. The quadratic optimal step size given in Eq. (**??**) gives the distance to the turning point of the error along the search direction under the quadratic assumption. This turning point nay be a maximum or a minimum. The Hessian can be used to find whether the turning point is a maximum or a minimum.

The product $H\mathbf{d}$ of the Hessian and a direction $\mathbf{d}$ is the rate of change of gradient in the direction $\mathbf{d}$. The expression $\mathbf{d}^T \; H \; \mathbf{d}$ is negative if the gradient is increasing along the direction $\mathbf{d}$, and positive if the gradient is decreasing along the direction $\mathbf{d}$. Within the quadratic assumption, the Hessian is constant, and the sign of the change of gradient along the direction is constant. If the gradient is increasing, the graph of the error along $\mathbf{d}$ is a cup, and there is a minimum, otherwise there is a maximum.

To monitor the sign of the product $\mathbf{d}^T \; H \; \mathbf{d}$, and therefore the type of the turning point, define δ by $\delta = \mathbf{d}^T \; H \; \mathbf{d}$. If $\delta \leq 0$ for non-zero $\mathbf{d}$, there is a minimum along the direction $\mathbf{d}$. But for non-quadratic error surface, it may be that $\delta \leq 0$ and yet there is still a minimum to find, since H changes along $\mathbf{d}$ [?]. The SCG method takes account of this. *Mòller* introduces two new variables, λ and $\bar{\lambda}$, to define an altered value of δ and $\bar{\delta}$. These variables are charged with ensuring that $\bar{\delta} > 0$. $\bar{\delta}$ is defined as follows:

$$\bar{\delta} = \delta + (\bar{\lambda} - \lambda)\mathbf{d}^T\mathbf{d}. \tag{6}$$

The requirement for $\bar{\delta} > 0$ gives a condition for $\bar{\lambda}$:

$$\bar{\lambda} > \lambda - \frac{\delta}{\mathbf{d}^T\mathbf{d}}. \tag{7}$$

Mòller then sets $\bar{\lambda} = 2(\frac{\delta}{\mathbf{d}^T\mathbf{d}})$ to satisfy Eq. (**??**) and so ensures $\bar{\delta} > 0$. This allows for a substitution in Eq. (**??**) to give:

$$\bar{\delta} = -\delta + \lambda\ \mathbf{d}^T\mathbf{d}. \tag{8}$$

Subsequently, $\bar{\delta}$ is substituted for expressions that would otherwise involve δ. Thus the step size η is found by substituting $\bar{\delta}$ for δ in Eq. (**??**):

$$\eta_i = \frac{-\mathbf{d}_i^T \nabla E_i}{\bar{\delta}} = \frac{\mathbf{d}_i^T \nabla E_i}{\bar{\delta} - \lambda \mathbf{d}_i^T\ \mathbf{d}_i}. \tag{9}$$

It is now necessary to specify how to update the scale λ. This is based on a measure of fit between the quadratic approximation and the real surface. The fit is measured by the variable Δ, and is a ratio between the actual change in error E produced by stepping along the search direction $\mathbf{d}$ by an amount η, and the predicted change in that error based on the quadratic approximation. λ is then updated as follows:

$$\lambda_{i+1} = \begin{cases} \frac{1}{4}\lambda_i, & if \quad \Delta_i > 0.75 \\ \lambda_i + \frac{\mathbf{d}_i^T\ H_i\ \mathbf{d}_i(1-\Delta_i)}{\mathbf{d}_i^T\ \mathbf{d}_i}, & if \quad 0 < \Delta_i < 0.25 \\ \lambda_i & otherwise \end{cases} \tag{10}$$

where λ_i is the value of λ at iteration i.

Class	Substance	Training	Validation
1	1C	68	82
2	1E	59	91
3	1F	74	77
4	1L	67	83
5	1P	82	68
6	1U	72	78
7	2C	71	79
8	2E	91	66
9	2F	77	73
10	2L	73	77
11	2P	77	73
12	2U	68	82
Total	1808	879	929

Table 1: List of substances and the number of spectra used for training/validation: C (control); E (ethephon); F(foil); L (leaf-removal); P (plastic-house); U (unpruned), and prefix 1 and 2 indicate harvest and after storage time, respectively.

3 Experimental

3.1 Data collection and Pre-processing

For the spectral data, visible-NIR spectra were collected at harvest and after storage at 0° C for 16 weeks from kiwifruit berries sourced from six pre-harvest fruit management treatments applied to vines grown at the *Blands Research Orchard* near *Hamilton, New Zealand* in the 1996 harvest season. The treatments were as follows:

Control: Normal thinning of annual vegetative growth;

Ethephon: Thinning of annual vegetative growth, dipping the berries in a ripening agent (7 mg / L ethephon) two weeks prior to harvest set with normal thinning of annual vegetative growth;

Foil: Wrapping the berries in foil to exclude light at one month after fruit-set, together with normal thinning of annual vegetative growth;

Leaf-Removal: Removing of leaves and fruiting shoots two months prior to harvest, together with normal thinning of annual vegetative growth;

Plastic-House: Covering the vine with plastic film, with no thinning of annual vegetative growth;

Unpruned: No thinning of annual vegetative growth.

Diffuse reflectance spectra (consisting of 1024 averaged measurements over the 516-998nm range) were collected with a fibre optic probe visible-near infrared spectrometer (PS1000 Ocean Optics Inc. Florida) from a single equatorial location on each berry using 150 berries per treatment. Reference spectra were collected from a white halon tile and spectral data were corrected for baseline drift where it occurred. The reflectance spectra were calculated as the ratios of the signal to the tile reference at each wavelength. The first 11 CV scores were calculated for each fruit from the PC, FFT, FHT and HT transforms of these reflectance spectra as described in Table **??**.

The primary feature extraction techniques used were PC, FFT, FHT, RR (Hurst), Haar and polar coordinate (calculated centroid for mean centred log(1/R) data) transformations. Canonical variates (cv) were calculated for the FFT, FHT, Haar, RR and PC feature data sets, thus deriving a secondary set of 11 data per fruit. This latter set, or the original extracted features were used to evaluate their suitability for categorising the fruit treatments by the pattern recognition techniques.

A short description on the features and data sets transformed by canonical variate analysis are listed in Table **??**. Table **??** summarised the substance name and the number of spectra used in learning and testing. The performance of the classification algorithms was assessed using both the primary extracted features, or the CV scores derived from them. In either case the data was divided into training and validation subsets comprising 879 and 929 spectra, respectively.

Fig. **??** shows the scores of the first two canonical variates (cv) of the validation sets for the spectra taken at harvest and after storage. The berries at harvest and after storage were segregated clearly, although the foil shaded and ethephon treated berries were clearly distinguished at harvest, but these were not at later time.

3.2 Software and hardware used

The features were extracted using Array Basic scripts running under Grams 32 software (Galactic industries Corp., Salem, NH, USA). JMP (SAS Institute, Cary, NC, USA) was used for the CV and LD analyses. The experiments were carried

Table 2: Extracted Spectra features used to pre-process spectra

Group	Feature	Short Remarks
1	FFT	The fast Fourier transform (FFT) coefficients (50 per fruit)
	FHT	The fast Hartley transform (FHT) coefficients (49 per fruit)
	Haar	The 4th order Haar transform coefficients (60 per fruit)
	RR(Hurst)	The rescaled range (log(range/SD)) (13 per fruit)
	PC	The principal component (pc) scores (14 per fruit)
	Polar	The polar coordinates of point and surface calculated centroid for mean centred log(1/R) data [a]
2	cv-FFT	The canonical variate (cv) scores (11) derived from the FFT data and knowledge of the treatments
	cv-FHT	The canonical variate (cv) scores (11) derived from the FHT data and knowledge of the treatments
	cv-Haar	The canonical variate (cv) scores (11) derived from the Haar transform data and knowledge of the treatments
	cv-Hurst	The canonical variate (cv) scores (11) derived from the Hurst Range Renormalisation (RR)
	cv-PC	The canonical variate (cv) scores (11) derived from the pc data and knowledge of the treatments

[a] Polar coordinates provide a 2D map of the higher dimensional space of the 512 point spectra, the point and surface centroids are 2 different methods for calculating the moment of the shape that the spectra take when plotted as polar coordinates. X_p and Y_p describe the centroid of a fruit spectra derived from the points, and X_s and Y_s describe the centroid of a fruit spectra derived from the surface occupied by the spectra.

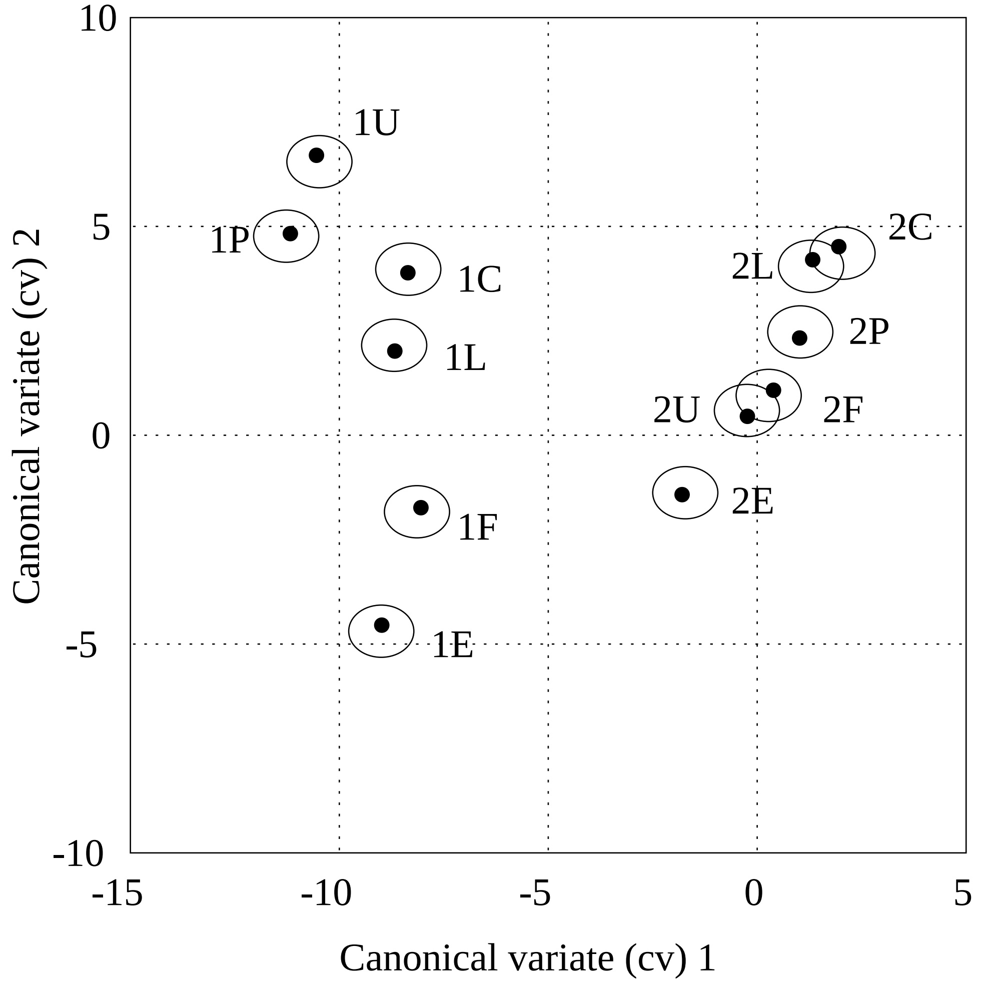

Figure 3: Mean scores and approximate 99 % confidence circles for the first two canonical variates (cv) derived from visible-NIR spectra of treated kiwifruit: control (c), ethephon (e), foil (f) and leaf removal (l) treatments plus unpruned (u) and plastic-house (p) treatments at harvest (1) and after storage (2)

out using variously a SUN Sparcstation (SunOS 5.6) and a PC (Microsoft Windows NT 4.0). The programs for the simulation of the MLPs with variations of BP were written in *MATLAB* scripts and *MATLAB* NN Toolbox (The MathWorks Inc., MA).

3.3 Network topologies

In this study, the MLPs are constructed of three layers. We define a feature vector $\mathbf{F} = (f_1, f_2, \ldots, f_N)$ composed of various number of features which are listed in Table **??**; its task is to produce the correct output label for each input feature vector. The architecture of our supervised network is a MLP with 1 hidden layer.

An important question is what kind of output encoding to use for the labels. We use a local 1-out-of-N code in which each unit represents a different label: we assume that we have the classification information available for the training set, so added 20 components to the training data, corresponding to an 1-of-N binary coding of the groups. This code has the advantages that the trained classifier can be decomposed very simply into N separate classifiers and, furthermore, the outputs have a simple interpretation as approximations to the *a posteriori* probabilities [**?**]. The classification is determined by the maximum output value and we have selected an approach in which a class label is assigned to a case when one and only one output node has an output that exceeds a node-dependent threshold. In this study we have set the threshold for all nodes at 0.5, the range of possible outputs being $0 \sim 1$. When this requirement is not fulfilled, the case stays unclassified.

Before training, all networks were initialised by random numbers drawn from a normal distribution with zero mean and a standard deviation of 0.01. Thus, all algorithms had equal initial conditions. In Rprop, SCG and ABP, the number of neurons in the hidden layer ranged from 5 to 15 while the output layer had

12. The optimal network sizes and parameters used for each of the problems are listed in Tables **??**, **??** and **??**.

There are three parameters of interest in ABP: the learning rate (η); the increasing learning rate (η^+); and the decreasing learning rate (η^-). The learning rate(η) was set to various factors depending on the problems, η^+ to 1.05, and η^- to 0.7. And there are one parameter of interest in SCG: the regulating parameter (λ). The regulating parameter (λ) of the Hessian was set to different values depending on the problems. On the other hand, Rprop used an initial step width ($\eta(0)$) of 0.01, up/down factors (Δ^+/Δ^-) of 1.2/0.5 and a minimal/maximal step width ($\Delta_{min}/\Delta_{max}$) of 0.07/50.0, respectively. The optimal network sizes of different tasks are described in Tables **??**, **??** and **??**.

4 Results and Discussion

For each of the problems, an appropriately sized network was constructed as in Tables **??**, **??**, and **??**. The input layer contained as many neurons as the number of dimensions of the data set. The output layer contained as many neurons as the number of classes present in the data. Since the input and output of the network are fixed by the problem, the only layer whose size had to be determined is the hidden layer. Also, since we had no *a priori* information on how the various input characteristics affect the classification, we chose not to impose any structure on the connection patterns in the network. Our networks were thus fully connected.

There have been several heuristics proposed to determine an appropriate number of hidden-layer nodes. It should be small enough to permit generalisation from the training data and large enough to form an adequate internal representation of the domain. A good heuristic that we utilised was to set the number of hidden-layer nodes to be a fraction of the number of features taking care that it does not significantly exceed the number of classes in the domain.

Each method previously discussed was operated with a wide range of the parameters that control its behaviour. Many techniques exist for solving the problem of finding optimal values for the network parameters. We trained with five choice of the control parameters and the choice leading to the best performance was considered for performance evaluation. Each network was trained until the weight converged, i.e., when subsequent iterations did not cause any significant changes to the weight vector. The optimal network is assumed to be the simplest network which achieves the minimum error on the test set. From the available data, we determine that the network with 10 hidden units is the optimum in almost all different set of features. We report the results from only the best set of parameters and due to space considerations, we provide only the generalisation accuracy.

We wish to analyse the performance of the neural network classifiers in order to determine its limitations. It is well-known that the leave-one-out method of error rate estimation is preferable to the optimistically biased re-substitution method. However, in our case, leave-one-out would be impractical due to the computational complexity for re-training the network many times. We therefore use the hold-out method and divide the available labelled examples randomly into a training set and a test set.

In each of the techniques, the number of patterns classified correctly was determined as follows: we consider first the confusabilities of the labels. A standard analysis technique from statistical pattern recognition theory is to compute the confusion matrix which is defined as:

$$C_{ij} = \sum_{k=1}^{N} \hat{\omega}_i(x_k)\omega_j(x_k), \tag{11}$$

where $\hat{\omega}_i(x_k)$ is the true class of pattern x_k and $\omega_j(x_k)$ is the class predicted by the classifier. The classification accuracy for a true pattern class $\hat{\omega}_i$ and a predicted class ω_j is defined as:

$$A_i = \mathcal{P}(\omega_i|\hat{\omega}_i)\mathcal{P}(\hat{\omega}_i) \tag{12}$$

from which obtains the expected classification accuracy:

$$A = \mathcal{E}[A_i] = \frac{1}{M}\sum_{i=1}^{M}\mathcal{P}(\omega_i|\hat{\omega}_i)\mathcal{P}(\hat{\omega}_i) \tag{13}$$

A picture of the confusion matrix C_{ij} (true classes across rows, predicted classes across columns) emphasises the performance more clearly as a concentration of values on the diagonal.

4.1 Neural Network Classifiers

MLP networks perform a mapping from the problem characteristic vector to an output vector describing class memberships. The generalisation performance of the network is measured by the classification accuracy on a validation set of unseen examples. The best possible performance is that theoretically achievable by the Bayes classifier. However, it is known that ANNs of the MLP architecture are able to approximate the optimal Bayes decision boundaries [**?**].

The training set consists of 879 measurement vectors and again 929 measurements are available for testing. The training period was limited to 3000 epochs using a fixed 3 layer network architecture with 12 output units and the network was fully connected. The weights of the network have been randomly chosen by a normal distribution ($\mu = 0.0, \sigma = 0.01$). Experiments to obtain the best combination of parameters have been carried out. Training stops when any of the following conditions occur:

- The maximum number of epochs (steps) is reached.
- Percentage of correctly classified cases greater than some threshold.
- Performance (total error) has been minimised to the goal (0.0001).

The resulting classifier is tested with fresh data to evaluate the goodness of the model. Rather than adapting the weights after processing of each case (on-line or incremental mode), the weights in the network were updated after each complete pass through all learning cases (batch mode).

SCG assumed a low value of the regulating factor (λ), approximately 5×10^7. Also, the parameter influences the performance of SCG. The parameters such as the learning rate (η), the increasing learning rate (η^+), and the decreasing learning rate (η^-) were chosen a fixed value of 0.01, 1.05, and 0.7, respectively. Rprop also provided a similar power of performance, but had an extremely fast convergence rate. We chose a fixed value of η_{min} because the algorithm refines it iteratively and we set an upper bound 50 on the weight changes Δ_{max}. The best performance were achieved at $(\Delta_{min}, \Delta_{max}) = (0.07, 50)$. It was observed that the ideal value of Δ_0 was in the range, $0.01 \sim 0.07$.

See Tables **??**, **??**, and **??** for the performance of supervised neural networks on the 11 data sets. Networks with very good performance have been trained. The performance measurement of the mean square error (MSE) and the mean prediction error (MPE) are used for training and validation, respectively. The three paradigms classify most of the problem patterns correctly, although the Hurst and Polar transformed features gave a high proportion ($> 50\%$) of misclassifications. The generalisation error on the Group 1 data sets resulted in an extremely low error for SCG, and, conversely, a little bit higher error for the Rprop and ABP algorithms. SCG was found to be a very good algorithm for classification in Haar transformed data set. However, training with the larger network with the smaller training set instead of the smaller network leads to the expected degradation of performance.

Features	Network	Parameter (Δ_0)	MSE	MPE	accuracy (%)
FFT	50-10-12	0.07	0.0035	0.0085	92.36
FHT	49-10-12	0.07	0.0058	0.0109	88.48
Haar	60-10-12	1×10^{-6}	0.016	0.0130	86.33
RR(Hurst)	13-10-12	0.01	0.044	0.073	49.84
PC	14-9-12	0.07	0.0039	0.0092	88.70
Polar	4-9-12	0.07	0.061	0.099	35.85
cv-FFT	11-9-12	0.01	0.0021	0.0076	89.24
cv-FHT	11-10-12	0.01	0.003	0.0274	82.99
cv-Haar	11-9-12	0.07	0.0047	0.0189	84.92
cv-Hurst	11-10-12	0.01	0.041	0.0889	46.72
cv-PC	11-15-10	0.07	0.0048	0.0097	87.62

Table 3: Results for Resilient back-propagation (Rprop)

Features	Network	Parameter (λ)	MSE	MPE	accuracy (%)
FFT	50-10-12	1×10^{-5}	9×10^{-6}	0.0105	93.65
FHT	49-10-12	1×10^{-6}	0.00099	0.00254	86.98
Haar	60-10-12	5×10^{-7}	0.00029	0.00013	93.76
RR(Hurst)	13-10-12	0.001	0.039	0.0539	46.39
PC	14-11-12	5×10^{-7}	0.0013	0.0042	86.98
Polar	4-10-12	5×10^{-7}	0.062	0.0929	37.03
cv-FFT	11-5-12	1×10^{-6}	0.0067	0.0076	87.19
cv-FHT	11-15-12	1×10^{-6}	0.0014	0.00574	81.05
cv-Haar	11-7-12	5×10^{-6}	0.0053	0.00509	82.13
cv-Hurst	11-10-12	0.1	0.039	0.0889	44.03
cv-PC	11-9-12	5×10^{-6}	0.0044	0.0057	86.11

Table 4: Results for Scaled Conjugate Gradient (SCG)

Features	Network	Parameter (η)	MSE	MPE	accuracy (%)
FFT	50-10-12	0.01	0.0048	0.0055	93.0
FHT	49-15-12	0.01	0.011	0.034	87.73
Haar	60-15-12	0.01	0.025	0.043	83.42
RR(Hurst)	13-15-12	0.01	0.046	0.079	47.15
PC	14-15-12	0.01	0.0016	0.0042	87.94
Polar	4-15-12	0.01	0.064	0.103	34.34
cv-FFT	11-15-12	0.01	0.0083	0.0086	87.84
cv-FHT	11-15-12	0.01	0.0051	0.0094	82.35
cv-Haar	11-15-12	0.01	0.0019	0.0079	82.88
cv-Hurst	11-15-12	0.01	0.036	0.0589	42.84
cv-PC	11-15-12	0.01	0.0054	0.009	86.33

Table 5: Results for Adaptive Back-Propagation (ABP)

4.2 Analysis

Previous work had shown that, in general, linear discrimination analysis (LDA) models, using canonical variate (cv) scores derived from principal component (PC) scores of visible-NIR spectra, were adequate for distinguishing between the groups of kiwifruit berries [?]. However, the connectionist methods that we tried out performed well, in terms of % accuracy. The results are shown in Tables **??**, **??** and **??**. The tables list the predictive abilities of the three neural networks for both groups of benchmark tasks. According to classification performance connectionist methods are comparable to statistical models in terms of generalisation errors and accuracy.

In summary, the neural network algorithms appear to achieve high accuracies consistently for all data sets. Table **??** and Fig. **??** summarise the classification accuracies of these algorithms. For the MLP methods, FFT feature extraction gave consistently good classification, although the best classification overall was given by the Haar transform in conjunction with SCG.

The performance of the SCG and LDA classifiers for FFT and Haar transforms after applying the unseen examples is shown in the confusion matrices in Tables **??**, **??**, **??** and **??**. The best classification overall was given by SCG-Haar, followed closely the SCG-FFT where the misclassification was approximately one half that given by the linear model.

Overall the conversion of the features by forming the canonical variates gave inferior classification. The RR (or Hurst) and Polar transforms gave a high proportion ($> 50\%$) of misclassifications, although still significantly better than a random assignment. For this dataset, linear discrimination models based on the features extracted by FFT gave a superior classification in comparison to the features extracted by the other methods. However, the overall classification error based on the FFT based linear discrimination model was about 12%; about twice the error of the best MLP-based classifiers.

5 Conclusions

This work demonstrates that a classification system's performance depends strongly on the feature extraction used, and the subsequent selection methods, which is particularly true for the problems we considered here. This was achieved without recourse to other measurements, such as tactile firmness, which would exhibit more characteristic of components. While each and every orchard in New Zealand turns out a product which is unique and which has distinctive qualities, it would clearly be difficult if not possible to try them all out. In such cases, we show that artificial neural network or connectionist methods can successfully be used. These techniques build models directly based on process measurements, and thus pro-

Features	Multilayer Perceptrons			DA
	Rprop	SCG	ABP	
FFT	7.64	6.35	7.0	11.84
FHT	11.52	13.02	12.27	19.81
Haar	13.67	6.24	16.58	14.32
RR (Hurst)	50.16	53.61	52.85	56.08
PC	11.3	13.02	12.06	20.34
Polar	64.15	62.97	65.66	71.15
cv-FFT	10.76	12.81	12.16	11.09
cv-FHT	17.01	18.95	17.65	19.48
cv-Haar	15.18	17.87	17.12	13.89
cv-Hurst	53.28	55.97	57.16	56.08
cv-PC	12.38	13.89	13.67	17.55

Table 6: The overall classification performance (% error) of the discriminate analysis (DA) and connectionist algorithms

	1C	1E	1F	1L	1P	1U	2C	2E	2F	2L	2P	2U	total
1C	71	0	0	2	0	9	0	0	0	0	0	0	82
1E	0	89	4	0	0	0	0	0	0	0	0	0	93
1F	0	0	69	0	0	0	0	0	0	0	0	0	69
1L	9	0	1	79	1	0	0	0	0	0	0	0	90
1P	1	0	2	2	59	4	0	0	0	0	0	0	68
1U	1	0	0	0	8	65	0	0	0	0	0	0	74
2C	0	0	0	0	0	0	59	0	0	15	0	0	74
2E	0	2	1	0	0	0	0	65	0	1	0	0	69
2F	0	0	0	0	0	0	0	0	68	0	12	3	83
2L	0	0	0	0	0	0	20	0	0	60	2	0	82
2P	0	0	0	0	0	0	0	0	3	1	58	0	64
2U	0	0	0	0	0	0	0	1	2	0	1	77	81
total	82	91	77	83	68	78	79	66	73	77	73	82	929

Table 7: Confusion matrix for the linear discrimination of kiwifruit treatments based on features extracted by FFT of the validation set of VNIR spectra (Error: 11.84%)

	1C	1E	1F	1L	1P	1U	2C	2E	2F	2L	2P	2U	total
1C	73	0	0	5	1	11	0	0	0	0	0	1	91
1E	0	86	1	0	0	0	0	0	0	0	0	0	87
1F	0	1	74	0	0	0	0	0	0	0	0	0	75
1L	8	2	1	76	0	0	0	0	0	0	0	0	87
1P	1	0	0	2	56	3	0	0	0	0	0	0	62
1U	0	0	0	0	11	64	0	0	0	0	0	0	75
2C	0	0	0	0	0	0	53	0	0	13	0	0	66
2E	0	2	1	0	0	0	0	66	0	0	0	1	70
2F	0	0	0	0	0	0	0	0	62	0	15	3	80
2L	0	0	0	0	0	0	26	0	0	62	2	0	90
2P	0	0	0	0	0	0	0	0	4	2	52	5	63
2U	0	0	0	0	0	0	0	0	7	0	4	72	83
total	82	91	77	83	68	78	79	66	73	77	73	82	929

Table 8: Confusion matrix for the linear discrimination of kiwifruit treatments based on features extracted by Haar transformation of the validation set of VNIR spectra (Error: 14.32%)

	1C	1E	1F	1L	1P	1U	2C	2E	2F	2L	2P	2U	total
1C	74	0	0	0	0	8	0	0	0	0	0	0	82
1E	0	89	2	0	0	0	0	0	0	0	0	0	91
1F	0	1	75	0	0	0	0	0	0	0	0	0	76
1L	0	0	0	80	1	0	0	0	0	0	0	0	81
1P	1	0	0	3	62	4	0	0	0	0	0	0	70
1U	7	0	0	0	5	66	0	0	0	0	0	0	78
2C	0	0	0	0	0	0	73	0	0	2	0	0	75
2E	0	1	0	0	0	0	0	66	0	0	0	0	67
2F	0	0	0	0	0	0	0	0	67	0	3	6	76
2L	0	0	0	0	0	0	6	0	0	74	1	0	81
2P	0	0	0	0	0	0	0	0	1	1	68	0	70
2U	0	0	0	0	0	0	0	0	5	0	1	76	82
total	82	91	77	83	68	78	79	66	73	77	73	82	929

Table 9: Confusion matrix for the SCG of kiwifruit treatments based on features extracted by FFT of the validation set of VNIR spectra (Error: 6.35%)

	1C	1E	1F	1L	1P	1U	2C	2E	2F	2L	2P	2U	total
1C	76	0	0	0	0	7	0	0	0	0	0	0	83
1E	0	87	0	2	0	0	0	0	0	0	0	0	89
1F	0	0	76	1	1	0	0	0	0	0	0	0	78
1L	3	0	0	78	0	0	0	0	0	0	0	0	78
1P	3	0	0	2	59	3	0	0	0	0	0	0	67
1U	0	0	0	0	8	68	0	0	0	0	0	0	79
2C	0	0	0	0	0	0	71	0	0	0	0	0	71
2E	0	4	1	0	0	0	0	66	0	0	0	1	72
2F	0	0	0	0	0	0	0	0	66	0	2	2	70
2L	0	0	0	0	0	0	8	0	0	76	2	0	86
2P	0	0	0	0	0	0	0	0	5	1	69	0	75
2U	0	0	0	0	0	0	0	0	2	0	0	79	81
total	82	91	77	83	68	78	79	66	73	77	73	82	929

Table 10: Confusion matrix for the SCG of kiwifruit treatments based on features extracted by Haar transformation of the validation set of VNIR spectra (Error: 6.24%)

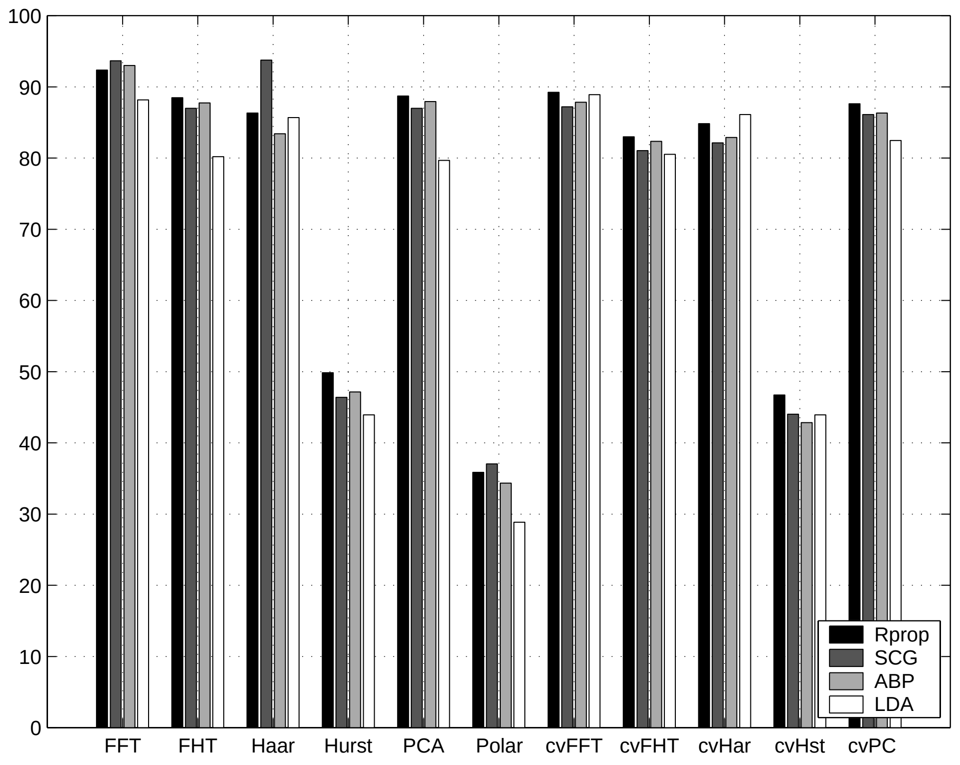

Figure 4: Results of classification: Comparison of Rprop, SCG, ABP and LDA

vide a means to analyse processes without explicit physical process model. The data sets have been transformed into various PC, FFT, FHT, Haar, Hurst transform, associated canonical variate (cv) transforms and Polar coordinate datasets. We also provide the insights to determine which features to select from the ones that we have extracted.

In order to evaluate the proposed task we used three variants of BP—Rprop, SCG and ABP as supervised neural networks and LDA as a linear model.

As it has been seen, the artificial neural network approach and the results obtained are rather different from the conventional model-based methods of problem solving. Neural networks are particularly interesting because of their inherent claim to analyse noisy or error data and to deal with problems that have no clear cut solution, in addition to their critical differentiation from other techniques, namely their ability to learn. This is important for those tasks where examples of the desired behaviour are easily given but where we are not sure how to arrive at the answer. It was shown that, if we have some idea of how to proceed, this can be used to form feature vectors which can be used to train a MLP.

In this study, supervised connectionist methods have proven to outperform the investigated conventional multivariate analysis and we guess that it is the fastest training algorithm at the time being. The general applicability of this result needs to be evaluated in different systems, but clearly show promise and may meet the requirements of practical fruit classification.

Although all of the methods gave some degree of classification, the high performance of the FFT extraction was unexpected in view of the current literature preference for PC [?], and clearly warrants more detailed investigation.

This work demonstrates that the classification performance depends strongly on the feature extraction used, and the subsequent selection methods. Artificial neural networks appear to be well suited to the classification of fruit grown or stored under different conditions. In particular this study has shown that non-linear classification based on the SCG algorithm could give low rates of error if combined with appropriate feature extraction methods. The classification was at a level of practical utility, and the approach may be more generally applicable to fruit classification.

Acknowledgements

The authors acknowledge the research grants (UOO808, HRO6806) funded by the Public Good Science Fund (PGSF) of the Foundation of Research Science and Technology (FRST) in New Zealand and would like to thank the reviewers for their valuable discussions and suggestions which have helped to improve the quality of this paper. The first author gratefully acknowledges the support from Research & International of the University of Otago under grant JDL-B15.

The main part of this study was done while the first author (JK) was working at the University of Otago, New Zealand.

References

[1] C. Borggaard and H.H. Thodberg, Anal. Chem., 64, 1992, pp. 545-551.

[2] R.N. Bracewell, Proceedings of the IEEE, 72(4), 1984, pp. 1010-1018.

[3] P. Cáceres-Alonso and A.J. García-Tejedor, J. Near Infrared Spectroscopy, 3, 1995, pp. 97-110.

[4] M.P. Cano, J. Agric. Food Chem., 39, 1991, pp. 1786-1791.

[5] L.W. Chan and F. Fallside, Computer Speech and Language, 2, 1987, pp. 205-218.

[6] C. Darken and J. Moody, In Neural Information Processing Systems, R.P. Lippmann, J.E. Moody and D.S. Touretzky (Eds.), 1991, pp. 832-838.

[7] I. Daubechies, IEET Trans. Inform. Theory, 36(5), 1990, pp. 961-1005.

[8] M. Defernez and R.H. Wilson, J. Sci. Food Agric., 67, 1995, pp. 461-467.

[9] G. Downey, J. Boussion, and D. Beauchene, J. Near Infrared Spectroscopy, 2, 1994, pp. 85-92.

[10] D.G. Evans, C.N.G. Scotter, L.Z. Day, and M.N. Hall, J. Near Infrared Spectroscopy, 1, 1993, pp. 33-44.

[11] A. Economou, P.R. Fielden, and A.J. Packham, The Analyst, 121, 1996, pp. 1015-1018.

[12] S.E. Fahlman, *In* Proc. 1988 Connectionist Models Summer School, T.J. Sejnowski, G.E. Hinton, and D.S. Touretzky (Eds.), San Mateo, CA: Morgan Kaufmann, 1988, pp. 38-51.

[13] R. Fletcher, John Wiley & Sons, NY: Academic Press Inc., 1975.

[14] I.M. De La Fuente, L. Martinez, J.M. Aguirregabiria, J. Veguillas, Fractals, 6, 1998, pp. 95-100.

[15] P. Geladi and E. Dåbakk, J. Near Infrared Spectroscopy, 3, 1995, pp. 119-132.

[16] J. Guthrie and K. Walsh, Austral. J. Exper. Agric., 37, 1997, pp. 253-263.

[17] A. Haar, Math. Ann., 69, 1910, pp. 331-373.

[18] J.B. Hampshire and B.A. Pearlmutter, *In* Proc. 1990 Connectionist Models Summer School, T.J. Sejnowski, G.E. Hinton, J.L. Elman and D.S. Touretzky (Eds.), San Mateo, CA: Morgan Kaufmann, 1990, pp. 159-172.

[19] M. Hana, W.F. McClure, and T.B. Whitaker, J. Near Infrared Spectroscopy, 3, 1995, pp. 133-142.

[20] S. Haykin, New York: MacMillian and IEEE Computer Society, 1994.

[21] J. Hertz, A. Krogh, and R.G. Palmer, Addison-Wesley, 1991.

[22] C. Hervás, A. Garrido, B. Lucena, N. García, and E. De Pedro, J. Near Infrared Spectroscopy, 2, 1994, pp. 177-184.

[23] M. Hestenes, Springer-Verlag, NY, 1990.

[24] K. Hornik, M. Stinchcombe, and H. White, Neural Networks, 2, 1989, pp. 359-366.

[25] P.J. Jackson and F.R. Harker, NZ J. Crop Hort. Sci., 25, 1997, pp. 185-189.

[26] R.A. Jacobs, Neural Networks, 1, 1988, pp. 295-307.

[27] W.J. Jasper and E.T. Kovacs, Textile Res. J., 64(8), 1994, pp. 444-448.

[28] R.B. Jordan, S.D. Osborne, R. Kunnemeyer, and R.J. Seelye, Proc. from the Sensors for Non-destructive Testing, Orlando, Florida, 1997, pp. 101-110.

[29] K.J. Kaffka and L.S. Gyarmati, J. Near Infrared Spectroscopy, 6, 1998, pp. A191-A200.

[30] N. Kasabov, The MIT Press: CA, MA, 1996.

[31] S. Kawano, H. Watanabe, and M. Iwamoto, J. Japan Soc. Hort. Sci., 61, 1992, pp. 445-451.

[32] J. Kim, A. Mowat, P. Poole, and N. Kasabov, Proc. Int. Conf. on Neural Information Processing (ICONIP'97), Dunedin, New Zealand, 1997, pp. 780-784.

[33] D. Lowe and A.R. Webb, IEEE Trans. Patt. Anal. and Mach. Intell., 13(4), 1991, pp. 355-364.

[34] J.R. Long, V.G. Gregoriou, and P.J. Gemperline, Anal. Chem., 62(17), 1990, pp. 1792-1797.

[35] M.F. $M\phi\ ller$, Neural Networks, 6(3), 1993, pp. 525-533.

[36] A.D. Mowat and P.R. Poole, Acta Horticulture, 436, 1997, pp. 159-163.

[37] A.D. Mowat and P.R. Poole, J. Near Infrared Spectroscopy, 5, 1997, pp. 113-122.

[38] T. Næs, K. Kvaal, T. Lsaksson, and C. Miller, J. Near Infrared Spectroscopy, 1, 1993, pp. 1-11.

[39] T. Onda, M. Tsuji, and Y. Komiyama, J. Japan Soc. Food Sci. Tech., 41, 1994, pp. 908-912.

[40] B.G. Osborne, B. Mertens, M. Thompson, and T. Fearn, J. Near Infrared Spectroscopy, 1, 1993, pp. 77-83.

[41] S.D. Osborne, R.B. Jordan, and R. Kunnemeyer, The Analyst, 122, 1997, pp. 15631-1537.

[42] S.D. Osborne, R. Kunnemeyer, and R.B. Jordan J. Near Infrared Spectroscopy, 7, 1999, pp. 9-15.

[43] M. Powell, Mathematical Programming, 12(2), 1977, pp. 241-254.

[44] B.A. Pearlmutter, Neural Computation, 6(1), 1994, pp. 147-160.

[45] M. Riedmiller and H. Braun, Proc. IEEE Int. Conf. on Neural Networks, 1993.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Nature, 323, October, 1986, pp. 533-536.

[47] T. Sato, J. Near Infrared Spectroscopy, 1, 1993, pp. 199-208.

[48] W. Schiffmann, M. Joost, and R. Werner, Proc. of the European Symposium on ANNs, ESANN'93, Brussels, 1993, pp. 97-104.

[49] J. Schmidhuber, North-Holland: Elsevier Science, 1989, pp. 439-445.

[50] F.M. Silva and L.B. Almeida, In Lecture Notes in Computer Science 412, L.B. Almeida and C.J. Wellkens (Eds.), Springer-Verlag, Berlin, 1990, pp. 110-119.

[51] E.W. Tollner, J.K. Brecht, and B.L. Upchurch, *In* Postharvest Handling: A Systems Approach, R.L. Shewfelt and S. Prussia (Eds.), Academic Press Inc., New York, 1992, pp. 43-71.

[52] T. Toolenaere, Neural Networks, 3, 1990, pp. 561-574.

[53] T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, and D.L. Alkon, Biol. Cybern., 59, 1988, pp. 257-263.

[54] U. Weigel and R. Herges, J. Chem. Inf. Comput. Sci., 32(6), 1992, pp. 723-731.

[55] P.J. Werbos, Ph.D. Thesis, Harvard University, Cambridge, MA, 1974.

[56] L.G. Weyer, and S.D. B, J. Near Infrared Spectroscopy, 4, 1996, pp. 163-174.

[57] H. Young, C.O. Perera, and V.J. Paterson, J. Sci. Food Agric., 58, 1992, pp. 519-522.

[58] J.M. Zurada, Boston: PWS, 1992.