Engineering Applications of Artificial Intelligence 79 (2019) 13–22

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Noise reduction for near-infrared spectroscopy data using extreme learning machines☆

Pablo A. Henríquez [a], Gonzalo A. Ruz [a,b,*]

[a] Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Av. Diagonal Las Torres 2640, Peñalolén, Santiago, Chile
[b] Center of Applied Ecology and Sustainability (CAPES), Santiago, Chile

ARTICLE INFO

Keywords:
Near-infrared spectroscopy
Parallel layers
Constrained optimization
Regression
Classification

ABSTRACT

The near infrared (NIR) spectra technique is an effective approach to predict chemical properties and it is typically applied in petrochemical, agricultural, medical, and environmental sectors. NIR spectra are usually of very high dimensions and contain huge amounts of information. Most of the information is irrelevant to the target problem and some is simply noise. Thus, it is not an easy task to discover the relationship between NIR spectra and the predictive variable. However, this kind of regression analysis is one of the main topics of machine learning. Thus machine learning techniques play a key role in NIR based analytical approaches. Pre-processing of NIR spectral data has become an integral part of chemometrics modeling. The objective of the pre-processing is to remove physical phenomena (noise) in the spectra in order to improve the regression or classification model. In this work, we propose to reduce the noise using extreme learning machines which have shown good predictive performances in regression applications as well as in large dataset classification tasks. For this, we use a novel algorithm called C-PL-ELM, which has an architecture in parallel based on a non-linear layer in parallel with another non-linear layer. Using the soft margin loss function concept, we incorporate two Lagrange multipliers with the objective of including the noise of spectral data. Six real-life dataset were analyzed to illustrate the performance of the developed models. The results for regression and classification problems confirm the advantages of using the proposed method in terms of root mean square error and accuracy.

1. Introduction

Near-infrared (NIR) (Pierna et al., 2011) spectroscopy are mainly used to measure light absorption of the so-called mid-infrared light, in order to identify and quantify various materials. Spectroscopy in combination with varied multivariate algorithms has played an important role for fast and nondestructive analysis in petrochemical, agricultural, medical, and environmental sectors (Liu et al., 2015a; Luypaert et al., 2007; Kim et al., 2010; Park et al., 2012).

According to the Beer–Lambert law, the absorption of light in a medium is proportional to the path length and the concentration of the absorbing agent. That is, there is a linear relationship between absorbance and concentration when the path length remains constant, which motivated the use of linear multivariate calibration techniques, such as multiple linear regression (MLR), principal components regression (PCR) (Keithley et al., 2009) and partial least squares regression (PLS) (Wilcox et al., 2016).

However, the linearity of the Beer–Lambert law is limited by chemical and instrumental factors, such as, deviations in absorptivity coefficients at high concentrations, non-symmetrical chemical equilibrium, intermolecular reactions, existence of humidity inducing hydrogen bonding, changes in temperature, non-monochromatic radiation, scattering of light, fluorescence or phosphorescence of the sample, stray light, nonlinear detector response (Despagne and Luc Massart, 1998), etc. When the system exhibits strong nonlinear behaviors, classical linear methods may not completely identify the relationship between the spectra and corresponding concentrations and thus would produce large errors in regression and classification problems. Many nonlinear techniques have been developed, such as, artificial neural network (ANN) (Hajnayeb et al., 2011), support vector machine (SVM) (Li et al., 2009), nonlinear partial least squares (Rosipal and Trejo, 2001), etc. These methods may perform well on nonlinear data but are computationally more complex than linear methods and have the limitation of being prone to overfitting (Peng et al., 2013).

In the case that the experimental measures deviate from the Lambert–Beer law, a suitable pre-processing must be considered to

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.engappai.2018.12.005.

* Corresponding author at: Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Av. Diagonal Las Torres 2640, Peñalolén, Santiago, Chile.
E-mail addresses: pabhenriquez@alumnos.uai.cl (P.A. Henríquez), gonzalo.ruz@uai.cl (G.A. Ruz).

https://doi.org/10.1016/j.engappai.2018.12.005
Received 27 November 2017; Received in revised form 23 October 2018; Accepted 17 December 2018
Available online 29 December 2018
0952-1976/

compensate for this nonlinear behavior. The disadvantage of including such additional factors is an increase in the complexity of the model and, in turn, it is likely to have a reduction of the robustness of the model for future predictions. All pre-processing techniques have the aim to reduce the noise in the data with the purpose of improving the characteristics looked for in the spectra. However, there is always the danger of choosing the wrong type or applying a pre-processing that is too severe that you may (unintentionally) delete valuable information. This problem is described in detail by Rinnan et al. (2009).

For near-infrared spectroscopy data, it generally contains linear and non-linear components in regression or classification problems. Linear methods may have difficulty obtaining a good performance, since the non-linearity is usually modeled in a limited way. However, the linear methods are more simple and stable. The non-linear methods can provide better performance than the linear methods, but are more complex. Therefore, a simple, fast, precise and effective method is required.

In the early 1990s, different authors (Schmidt et al., 1992; Pao et al., 1994) independently proposed feedforward neural networks comprising randomly initialized and untrained connections between the input layer and a hidden layer of non-linear neurons. Then, in the 2000s these type of networks were revisited under the name of Extreme Learning Machine (ELM) (Huang et al., 2004). Recently, a comparison of these neural networks with random weights, for classification and regression problems, was carried out by Henríquez and Ruz (2018a), also, in Zhang and Suganthan (2016b), a survey on randomized algorithms for training neural networks was presented. In this paper, we will use the term ELM as a reference for randomized feed-forward neural networks (Henríquez and Ruz, 2018b).

ELM has been successfully applied in many fields (Samat et al., 2014; Cao et al., 2016). Especially, for NIR data, ELM combined with feature selection techniques has been used to determine amino acid nitrogen in soy sauce (Ouyang et al., 2013), total acid content in vinegar (Chen et al., 2012), pear internal quality attributes (Jiang and Zhu, 2013), etc. Recently, in Li et al. (2016) was analyzed the feasibility of Fourier transform infrared transmission (FT-IR) spectroscopy to detect talcum powder illegally added in tea. In Yang and Sun (2016) the abilities of six popular multivariate classification techniques are compared, including ELM. In Bian et al. (2017) ELM was used for near-infrared spectral quantitative analysis of diesel fuel and edible blend of oil samples.

A main difficulty when working with NIR data is the fact that there are many pre-processing techniques from which to choose from (more details in Section 2), and even combinations of them, where different researchers use arbitrarily such methodologies, in order to achieve good performance in classification and prediction problems. With this motivation, there is a need for a robust methodology to solve this problem. Therefore, we propose to reduce the noise in NIR data by using a novel algorithm called C-PL-ELM, which has a parallel architecture. This algorithm has a non-linear layer in parallel with another non-linear layer, generating a more powerful nonlinear mapping. Using the soft margin loss function concept (Bennett and Mangasarian, 1992), we incorporate two Lagrange multipliers as optimization constraints (similar to the concept of support vector regression (Drucker et al., 1997)) with the objective of including the noise in spectroscopy data, thus avoiding the pre-processing of the experimental measures.

The remaining of the paper is organized as follows: In Section 2, we briefly review some of the most popular pre-processing techniques. The detail of the proposed C-PL-ELM algorithm is presented in Section 3. Simulation results and comparisons are provided in Section 4. Section 5 presents discussions on the performance of C-PL-ELM with respect to the different datasets. Conclusions are drawn in Section 6.

2. Brief review of Pre-processing techniques

The aim of signal pre-processing is to improve the data quality before modeling and to remove physical information from the spectra.

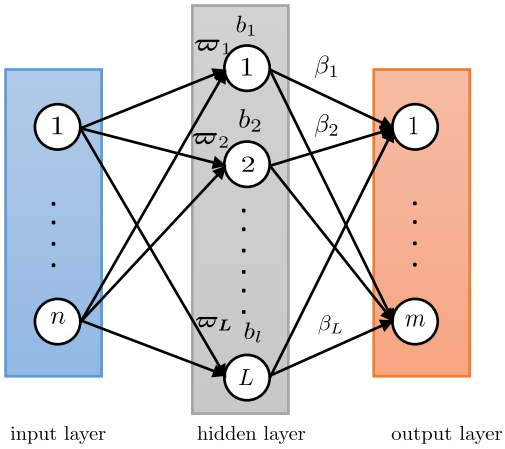

Fig. 1. Illustration of the ELM structure, which consists of an input layer, an output layer and a single hidden layer.

Applying pre-processing can increase the repeatability/reproducibility of the method, model robustness and accuracy, although there are no guarantees this will actually work.

The most widely used pre-processing techniques in NIR spectroscopy can be divided into two group: scatter-correction methods and spectral derivatives.

The first group of scatter-corrective pre-processing methods includes Multiplicative Scatter Correction (MSC), Inverse MSC, Extended MSC (EMSC), Extended Inverse MSC, de-trending, Standard Normal Variate (SNV) and normalization. The spectral derivation group is represented by: Norris-Williams (NW) derivatives and Savitzky-Golay (SG) polynomial derivative filters. The advantages and disadvantages of different preprocessing techniques are discussed in Rinnan et al. (2009).

Noise represents random fluctuations around the signal that can originate from the instrument or environmental laboratory conditions. The simplest solution to remove noise is to perform n repetition of the measurements, and then use the average spectra. The noise will decrease with a factor $\sqrt{n}$. When this is not possible, or if residual noise is still present in the data, the noise can be removed mathematically. In Table 1 we describe some commonly used mathematical models.

Therefore, the main goals of pre-processing in this context are:

- Reduce the noise in the spectrum.
- Improve exploratory analysis (e.g., using PCA).
- Improve classification or regression performance.

Different applications with NIR data and their respective pre-processing techniques are shown in Table 2.

3. Extreme learning machines

Extreme learning machine (ELM) (Huang et al., 2006b) is a unifying learning algorithm which can be used for several learning tasks. It was originally developed for the single hidden-layer feed forward neural networks (SLFNs), and then extended to the generalized SLFNs (Huang et al., 2012). In ELM, the feature mapping function is also called the activation function. The network structure of ELM is shown in Fig. 1. In accordance with the ELM's universal approximation capability theorems (Huang et al., 2006a), the activation function is required to be nonlinear piecewise continuous and various activation functions can be used. Several commonly used activation functions are listed in Table 3. In the second stage, the output weights are usually analytically determined by Moore–Penrose generalized inverse.

Table 1
Mathematical model of some pre-processing techniques. x_i or X is one original sample spectra measured by the NIR instrument, N is a normalizing coefficient, k is the number of neighbor values at each side of j and c_h are pre-computed coefficients, that depend on the chosen polynomial order and degree (smoothing, first and second derivate). X_c and X_s are the mean centered and auto-scaled matrices, $\overline{X}_j$ and s_j are the mean and standard deviation of variable j.

Name	Mathematical model
Savitzky-Golay filtering (Savitzky and Golay, 1964)	$x_j^* = \frac{1}{N}\sum_{h=-k}^{k} c_h x_{j+h}$
First derivative (Rinnan et al., 2009)	$x_i' = x_i - x_{i-1}$
Second derivative (Rinnan et al., 2009)	$x_i'' = x_{i-1} - 2x_i + x_{i+1}$
Standard Normal Variate (SNV)(Fearn, 2008)	$\mathrm{SNV}_i = \frac{x_i - \overline{x}_i}{s_i}$
Centering (Stevens and Ramirez-Lopez, 2013)	$Xc_{ij} = X_{ij} - \overline{X}_j$
Scaling (Stevens and Ramirez-Lopez, 2013)	$Xs_{ij} = \frac{X_{ij} - \overline{X}_j}{s_j}$
Multiplicative Scatter Correction (MSC) (Rinnan et al., 2009)	$x_i = b_0 + b_{ref,1}x_{ref} + e$ $\mathrm{MSC} = \frac{x_i - b_0}{b_{ref,1}} = x_{ref} + \frac{e}{b_{ref,1}}$

Table 2
Overview of the applications used with different pre-processing techniques.

Application	Spectral preprocess	Ref.
Detection of talcum powder in tea	Smoothing, normalized and SNV	(Li et al., 2016)
Determination of Protein Secondary Structure	Smoothing	(Wilcox et al., 2016)
Fescue grass powdered, pharmaceutical tablet, corn and meat	SNV	(Peng et al., 2013)
Determination of Amino Acid Nitrogen in Soy Sauce	First and second derivative, SNV and MSC	(Ouyang et al., 2013)
Total acid content (TAC) in vinegar	First and second derivative, SNV, MSC and smoothing	(Chen et al., 2012)
Determination of Pear Internal Quality Attributes	First and second derivative, SNV and MSC	(Jiang and Zhu, 2013)
Production of bioethanol from Pinus radiata pulps	Baseline correction	(del P. Castillo et al., 2015)
Super premium gasoline adulteration	First derivative, MSC, smoothing	(Mabood et al., 2017)
Predicting Soil Salinity	EPO and SNV	(Liu et al., 2015b)
Minced beef meat adulteration with turkey meat	First and second derivative, SNV, MSC and smoothing	(Alamprese et al., 2016)
Interactions of selenium species with living bacterial cells	Baseline correction and normalized	(Feo et al., 2004)
Quality of frozen guava and yellow passion fruit pulps	MSC, SNV, first derivative and smoothing	(Alamar et al., 2016)
Protein and wet gluten in commercial wheat flour	SNV, first and second derivative	(Chen et al., 2017)
Detection of fungal infections on citrus fruits	MSC and SNV	(Lorente et al., 2015)
Analyze powdered, pure and adulterated sweet potato	SNV	(Ding et al., 2015)
Detection of volatile compounds in apple wines	First and second derivative, MSC, normalization and ECO	(Ye et al., 2016)
Determination of egg content in dried egg-pasta	SNV	(Bevilacqua et al., 2013)
Detection of multiple adulterants in kudzu starch	First and second derivative, smoothing and SNV	(Xu et al., 2015)
Determination of total phenolic compounds in yerba mate	First derivative, MSC, and smoothing	(Frizon et al., 2015)

Table 3
Commonly used activation function in ELM.

Sigmoid function	$g(\boldsymbol{\varpi}, b, \mathbf{x}) = \frac{1}{1+\exp[-(\boldsymbol{\varpi}\cdot x+b)]}$
Gaussian function	$g(\boldsymbol{\varpi}, b, \mathbf{x}) = \exp(-b\|\mathbf{x} - \boldsymbol{\varpi}\|)$
Sine function	$g(\boldsymbol{\varpi}, b, \mathbf{x}) = \sin(\boldsymbol{\varpi}\cdot x + b)$
Cosine function	$g(\boldsymbol{\varpi}, b, \mathbf{x}) = \cos(\boldsymbol{\varpi}\cdot x + b)$
Hard limit function	$g(\boldsymbol{\varpi}, b, \mathbf{x}) = \begin{cases} 1, & \text{if } \boldsymbol{\varpi}\cdot x + b \geq 0 \\ 0, & \text{otherwise} \end{cases}$

3.1. Constrained-optimization-based ELM

Consider a set of N training examples $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N} \subset \Re^n \times \Re^m$ for the supervised learning process, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in \Re^n$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in \Re^m$ are the input vector and output vector, respectively, and the mathematical model of a SLFNs with L hidden nodes is described as

$$\hat{\mathbf{o}}_i = \sum_{j=1}^{L} \boldsymbol{\beta}_j g(\langle \boldsymbol{\varpi}_j, \mathbf{x}_i \rangle + b_j) \quad \forall i = 1, \ldots, N, \tag{1}$$

where $b_j \in \Re$ is the bias, g here represents the activation function of the network, $\boldsymbol{\beta}_j \in \Re^m$ is the link connecting the jth hidden node to the output node. $g(\langle \boldsymbol{\varpi}_j, \mathbf{x}_i \rangle + b_j)$ is the output of the jth hidden node with respect to the input sample $\mathbf{x}_i$, and $\langle \boldsymbol{\varpi}_j, \mathbf{x}_i \rangle = \boldsymbol{\varpi}\cdot x$ denotes the Euclidean inner product.

For all N samples, an equivalent compact form of (1) can be written as

$$\hat{\mathbf{O}} = \mathbf{H}\boldsymbol{\beta}, \tag{2}$$

where $H_{ij} = g(\langle \boldsymbol{\varpi}_j, \mathbf{x}_i \rangle + b_j)$ represents the entry in the ith row and jth column of the hidden layer output matrix $\mathbf{H}$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_L)$ and $\hat{\mathbf{O}} = (\hat{\mathbf{o}}_1, \hat{\mathbf{o}}_2, \ldots, \hat{\mathbf{o}}_N)$.

From (Huang et al., 2012), the constrained-optimization problem of regularized ELM is stated as (C-ELM)

$$\min_{\boldsymbol{\beta}, \xi_i} \mathcal{L} = \frac{1}{2}\|\boldsymbol{\beta}\| + \frac{C}{2}\sum_{i=1}^{N} \xi_i^2 \tag{3}$$

subject to: $\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{t}_i - \boldsymbol{\xi}_i, \ \forall i = 1, \ldots, N,$ (4)

where C and $\boldsymbol{\xi}$ are the cost parameter and slack variables of the Lagrange optimization function respectively. $\mathbf{t}_i$ is the corresponding target of the training data samples $\mathbf{x}_i$. The Lagrangian for this optimization problem is

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{\beta}\| + \frac{C}{2}\sum_{i=1}^{N} \xi_i^2 - \sum_{i=1}^{N} \boldsymbol{\alpha}_i(\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} - \mathbf{t}_i + \boldsymbol{\xi}_i), \tag{5}$$

where $\boldsymbol{\alpha}_i$ is the Lagrange Multiplier of the optimization function to be computed by the learning machines. Based on the Karush–Kuhn–Tucker (KKT) theorem the optimality conditions can be expressed as

$$\left.\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} &= 0\\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}_i} &= 0\\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}_i} &= 0\end{aligned}\right\}, \ \forall i = 1, \ldots, N. \tag{6}$$

By setting the gradient of $\mathcal{L}$ with respect to $\boldsymbol{\beta}$ to zero, we can get the closed-form solution of $\boldsymbol{\beta}$. There are two cases while solving $\boldsymbol{\beta}$. If $L < N$, the size of the matrix $\mathbf{H}^T\mathbf{H}$ is less than that of the matrix $\mathbf{H}\mathbf{H}^T$, then the closed-form solution can be obtained as

$$\boldsymbol{\beta} = \left(\mathbf{H}^T\mathbf{H} + \frac{I}{C}\right)^{-1}\mathbf{H}^T\mathbf{T}, \tag{7}$$

where I denotes the identity matrix of size L. If $L > N$, the size of the matrix $\mathbf{H}\mathbf{H}^T$ is less than that of the matrix $\mathbf{H}^T\mathbf{H}$, the solution of the equations is

$$\boldsymbol{\beta} = \mathbf{H}^T\left(\mathbf{H}\mathbf{H}^T + \frac{I}{C}\right)^{-1}\mathbf{T}. \tag{8}$$

For the multiclass case, the class label of a sample is formulated as

$$label(\mathbf{x}) = \arg \max_{1\le i\le m} \{\hat{\mathbf{o}}_i(\mathbf{x})\}. \tag{9}$$

3.2. Description of the proposed method

We propose to avoid the difficult and time consuming pre-processing stage for NIR data for regression and classification problems by directly incorporating in the training phase of the machine learning algorithm a parameter that can handle the noise, and therefore allowing the use of the NIR data without having to perform any of the pre-preprocessing techniques mentioned earlier. Given that ELM has shown promising results in this domain, we will adapt a recent version that considers a parallel architecture called PL-ELM which improves the predictive power of the classic ELM (Henríquez and Ruz, 2017b).

The proposed method incorporates two Lagrange multipliers that mimics support vector regression (SVR) (Drucker et al., 1997) into the basis of PL-ELM with the idea of mitigating the influence of the noise in the data samples. Our proposed method is called C-PL-ELM which is endowed with the ability to effectively deal with non-fixed and asymmetrical characteristics usually present in NIR data.

The C-PL-ELM algorithm (C-PL-ELM network structure is illustrated in Fig. 2) begins with the decision function as follows,

$$\hat{\mathbf{o}}_i = \sum_{j=1}^{L} \boldsymbol{\beta}_j \phi(\langle \mathbf{m}_j, \mathbf{x}_i\rangle) g(\langle \boldsymbol{\varpi}_j, \mathbf{x}_i\rangle + b_j), \tag{10}$$

where $\mathbf{m}_j$, $\boldsymbol{\varpi}_j$ are the weights matrix for each input layer, $\mathbf{x}_i$ the ith input sample, $\phi(\cdot)$ and $g(\cdot)$ are nonlinear activation function and b_j is the bias. In relation to the bias term in the output neurons, (Zhang and Suganthan, 2016a) analyzed empirically the effect of this term, concluding that the bias term in the output neurons has mixed effects on the performance, as it may or may not improve the performance. Therefore, from now on we will only use one bias.

Eq. (10) can be written compactly as

$$\hat{\mathbf{O}} = \mathbf{H}\boldsymbol{\beta}, \tag{11}$$

where

$$\mathbf{H} = \mathbf{H_1} \circ \mathbf{H_2}, \tag{12}$$

$$\mathbf{H_1} = \begin{bmatrix} \phi(\langle m_1, x_1\rangle) & \cdots & \phi(\langle m_L, x_1\rangle) \\ \vdots & \vdots & \vdots \\ \phi(\langle m_1, x_N\rangle) & \cdots & \phi(\langle m_L, x_N\rangle) \end{bmatrix} \in \Re^{N\times L}, \tag{13}$$

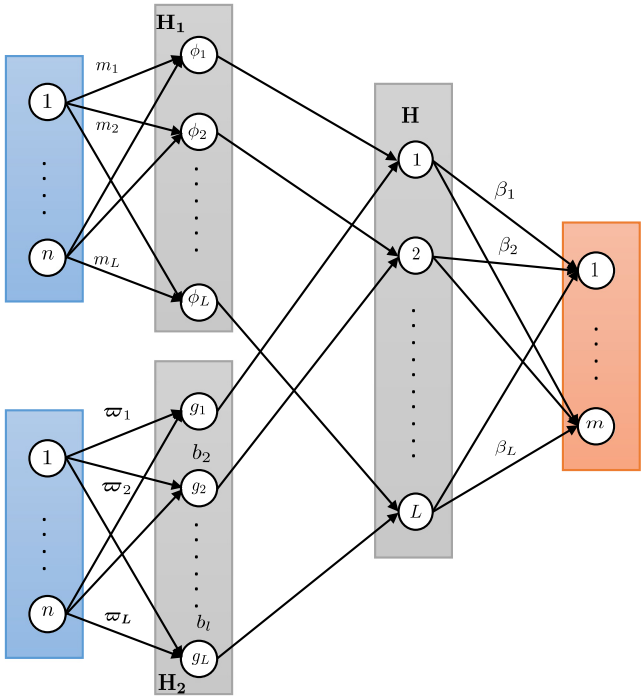

Fig. 2. Illustration of the C-PL-ELM structure. L hidden neurons for each parallel layer with two activation functions $\phi(\cdot)$ and $g(\cdot)$.

$$\mathbf{H_2} = \begin{bmatrix} g(\langle \boldsymbol{\varpi}_1, x_1\rangle + b_1) & \cdots & g(\langle \boldsymbol{\varpi}_L, x_1\rangle + b_L) \\ \vdots & \vdots & \vdots \\ g(\langle \boldsymbol{\varpi}_1, x_N\rangle + b_1) & \cdots & g(\langle \boldsymbol{\varpi}_L, x_N\rangle + b_L) \end{bmatrix} \in \Re^{N\times L}, \tag{14}$$

$\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_L)$ and $\hat{\mathbf{O}} = (\hat{\mathbf{o}}_1, \hat{\mathbf{o}}_2, \ldots, \hat{\mathbf{o}}_N)$. A least mean squares solution of (11) can be expressed by $\boldsymbol{\beta} = (\mathbf{H_1} \circ \mathbf{H_2})^{\dagger}\mathbf{O}$, where $(\mathbf{H_1} \circ \mathbf{H_2})^{\dagger}$ is the Moore–Penrose generalized inverse of the Hadamard product between $\mathbf{H_1}$ and $\mathbf{H_2}$. We can see from Eqs. (13) and (14) that each layer has the same number of hidden nodes L. The Hadamard product ($\mathbf{H} = \mathbf{H_1} \circ \mathbf{H_2}$) guarantees that the dimension of the $\mathbf{H}$ matrix will not change.

However, the least squares problem is usually ill-posed and one can employ a regularization model to find a solution, that is,

$$\min_{\boldsymbol{\beta}, \boldsymbol{\xi}_i} \mathcal{L} = \frac{1}{2}\|\boldsymbol{\beta}\| + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^{*2} \tag{15}$$

subject to: $\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = -\varepsilon + \mathbf{t}_i - \boldsymbol{\xi}_i, \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \varepsilon + \mathbf{t}_i + \boldsymbol{\xi}_i^*, \ \forall i = 1, \ldots, N.$

The soft margin loss function (Bennett and Mangasarian, 1992) which was used in support vector machines by Cortes and Vapnik (1995), one can introduce slack variables ($\boldsymbol{\xi}$, $\boldsymbol{\xi}^*$) to cope with, otherwise infeasible, constraints of the optimization problem. In (15), each training sample has its own corresponding ($\boldsymbol{\xi}$, $\boldsymbol{\xi}^*$) values which are used to determine whether the training instances fall outside the scope of ε. The penalty parameter $C > 0$ determines the trade-off between the flatness of the network output and the amount up to which deviations larger than ε are tolerated. This corresponds to dealing with a so called ε-insensitive loss function $|\boldsymbol{\xi}|_{\varepsilon}$ described by Smola and Schölkopf (2004):

$$|\boldsymbol{\xi}|_{\varepsilon} := \begin{cases} 0, & \text{if } |\boldsymbol{\xi}| \le \varepsilon, \\ |\boldsymbol{\xi}| - \varepsilon, & \text{otherwise.} \end{cases} \tag{16}$$

where ε for a Gaussian loss function has to be zero, i.e., $\varepsilon = 0$ (other loss functions are described in Smola and Schölkopf (2004)). We can transform the problem (15) into the solution of the dual problem and construct the following Lagrangian function

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\xi}, \boldsymbol{\xi}^*) = \frac{1}{2}\|\boldsymbol{\beta}\| + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^{*2} \tag{17}$$

$$-\sum_{i=1}^{N}\boldsymbol{\alpha}_i(\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta}-\mathbf{t}_i+\xi_i)-\sum_{i=1}^{N}\boldsymbol{\alpha}_i^*(-\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta}+\mathbf{t}_i+\xi_i^*),$$

$\mathbf{h}(\mathbf{x}_i)$ corresponds to the Hadamard product generated by each parallel layer of the network. $(\boldsymbol{\xi}, \boldsymbol{\xi}^*)$ are slack variables used to measure the error above and below the targeted output. $(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*)$ are the corresponding Lagrange multipliers. (17) was proposed for the classic version of the ELM in Wong et al. (2016) (This algorithm was called CO-ELM-R).

Using the well-known KKT theorem, we have

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\beta}}=0\rightarrow\boldsymbol{\beta}-\sum_{i=1}^{N}\boldsymbol{\alpha}_i\mathbf{h}(\mathbf{x}_i)+\sum_{i=1}^{N}\boldsymbol{\alpha}_i^*\mathbf{h}(\mathbf{x}_i)=0, \tag{18}$$

which yields,

$$\boldsymbol{\beta}=\mathbf{H}^T(\boldsymbol{\alpha}_i-\boldsymbol{\alpha}_i^*). \tag{19}$$

The derivatives of $\mathcal{L}$ with respect to the primal variables $(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\xi}, \boldsymbol{\xi}^*)$, can be rewritten as follows

$$\frac{\partial\mathcal{L}}{\partial\xi_i}=0\rightarrow C\xi_i-\boldsymbol{\alpha}_i=0, \tag{20}$$

$$\frac{\partial\mathcal{L}}{\partial\xi_i^*}=0\rightarrow C\xi_i^*-\boldsymbol{\alpha}_i^*=0, \tag{21}$$

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\alpha}_i}=0\rightarrow\xi_i-\mathbf{t}_i+\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta}=0, \tag{22}$$

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\alpha}_i^*}=0\rightarrow\xi_i^*+\mathbf{t}_i-\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta}=0. \tag{23}$$

By introducing (19) and (20) in (22), we obtain

$$\left(\mathbf{H}\mathbf{H}^T+\frac{I}{C}\right)\boldsymbol{\alpha}=\mathbf{T}+\mathbf{H}\mathbf{H}^T\boldsymbol{\alpha}^*, \tag{24}$$

in the same way, by introducing (19) and (21) in (23), we obtain

$$\boldsymbol{\alpha}^*=\left(\mathbf{H}\mathbf{H}^T+\frac{I}{C}\right)^{-1}\left(\mathbf{H}\mathbf{H}^T\boldsymbol{\alpha}-\mathbf{T}\right). \tag{25}$$

By introducing (25) in (24), we obtain

$$\boldsymbol{\alpha}=\left(2\mathbf{H}\mathbf{H}^T+\frac{I}{C}\right)^{-1}\mathbf{T}. \tag{26}$$

Also, by introducing (26) in (25), we obtain

$$\boldsymbol{\alpha}^*=-\left(2\mathbf{H}\mathbf{H}^T+\frac{I}{C}\right)^{-1}\mathbf{T}. \tag{27}$$

Using (12), we have the corresponding Lagrange multipliers

$$\boldsymbol{\alpha}=\left(2(\mathbf{H_1}\circ\mathbf{H_2})(\mathbf{H_1}\circ\mathbf{H_2})^T+\frac{I}{C}\right)^{-1}\mathbf{T} \tag{28}$$

$$\boldsymbol{\alpha}^*=-\left(2(\mathbf{H_1}\circ\mathbf{H_2})(\mathbf{H_1}\circ\mathbf{H_2})^T+\frac{I}{C}\right)^{-1}\mathbf{T}. \tag{29}$$

Finally, we can determine the slack variables using (28) and (29)

$$\xi_i=\frac{\boldsymbol{\alpha}}{C} \tag{30}$$

$$\xi_i^*=\frac{\boldsymbol{\alpha}^*}{C}. \tag{31}$$

3.2.1. Algorithm description

Given a training set with N points $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^N \subset \Re^n \times \Re^m$ and L hidden neurons in total with two activation functions $(\phi(\cdot), g(\cdot))$. Follow the next steps:

1. Randomly generate input weights using an uniform distribution $[-1, 1]$ ($\mathbf{m}_j$, $\boldsymbol{\varpi}_j$ and biases b_j)
2. Calculate the first parallel layer matrix $\mathbf{H_1}$ using (13).
3. Calculate the second parallel layer matrix $\mathbf{H_2}$ using (14).
4. Compute hidden layer output matrix, $\mathbf{H}=\mathbf{H_1}\circ H_2\subset\Re^{N\times L}$.
5. Compute Lagrange multipliers using (28) and (29).
6. Calculate output weight matrix $\boldsymbol{\beta}=\mathbf{H}^T(\boldsymbol{\alpha}_i-\boldsymbol{\alpha}_i^*)$.

4. Performance evaluation

In this section, the performance of the proposed C-PL-ELM learning algorithm is measured. All the simulations are carried out using the free R software for statistical computing environment running on a 2.6 GHz Intel Core i5 and 8 GB-RAM computer. In relation to the scope of the random weights and biases, the random values are considered typically in the range $[-1, 1]$ (as we do in this paper), some additional conditions are considered in Gorban et al. (2016); Li and Wang (2017); Tyukin and Prokhorov (2009); Henríquez and Ruz (2017a). All the experiment runs were performed 20 times and averages and standard deviations are reported. In all the simulations, the input and output variables of the two regression problems are normalized into the range $[0, 1]$, and the input variables of the four classification problems are normalized into the range $[-1, 1]$.

4.1. Data descriptions

Six spectral datasets were chosen for this study:

Dataset 1 contains a collection of 983 Mid-infrared spectra collected from different authenticated fruit purees in one of two classes: Strawberry and non-Strawberry (Holland et al., 1998).

Dataset 2 contains 120 spectra of fresh minced meats: chicken, pork and turkey. Duplicate acquisitions from 60 independent samples (Al-Jowder et al., 1997).

Dataset 3 contains coffee samples (Briandet et al., 1996) obtained by Fourier transform infrared spectroscopy with diffuse reflectance sampling. It contains 56 samples (arabica and robusta species, respectively 29 and 27 of each).

Dataset 4 consists of an olive oil dataset obtained using Fourier transform infrared spectroscopy with attenuated total reflectance sampling. Duplicate acquisitions from 60 authenticated extra virgin olive oils from 4 different countries of origin: Greece, Italy, Portugal and Spain (Tapp et al., 2003).

Dataset 5 contains near infrared measurements on brick cheese samples. Also included as independent variables are two temperature terms. It contains 140 samples and 16 attributes (14 independent and 2 dependent).[1]

Dataset 6 contains spectra of hydrocarbon mixtures from two different diode array spectrometers. Absorbances from 470 to 1100 nm were collected. It contains 60 samples and 320 attributes (316 independent and 3 dependent).[1]

In Fig. 3 we show the dataset 1, we can see the spectra without pre-processing and with pre-processing. From the same figure we can see the changes that the spectrum suffers due to different pre-processing techniques. As we have already described in Sections 1 and 2, the performance of the algorithms in classification and prediction problems is strongly influenced by the pre-processing used.

4.2. Model performance evaluation

In this paper, root mean square error (RMSE) and accuracy are used to measure the performance. The RMSE is defined as:

$$\text{RMSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_p(i)-f_o(i))^2} \tag{32}$$

where $f_p(i)$ and $f_o(i)$ are the predicted value and the target value, respectively. The accuracy is defined as:

$$\text{Accuracy}=\frac{\#(\text{correct classifications})}{\#(\text{all classifications})}. \tag{33}$$

[1] Dataset 5 and 6 were obtained from the Software Pirouette; Infometrix, Inc.

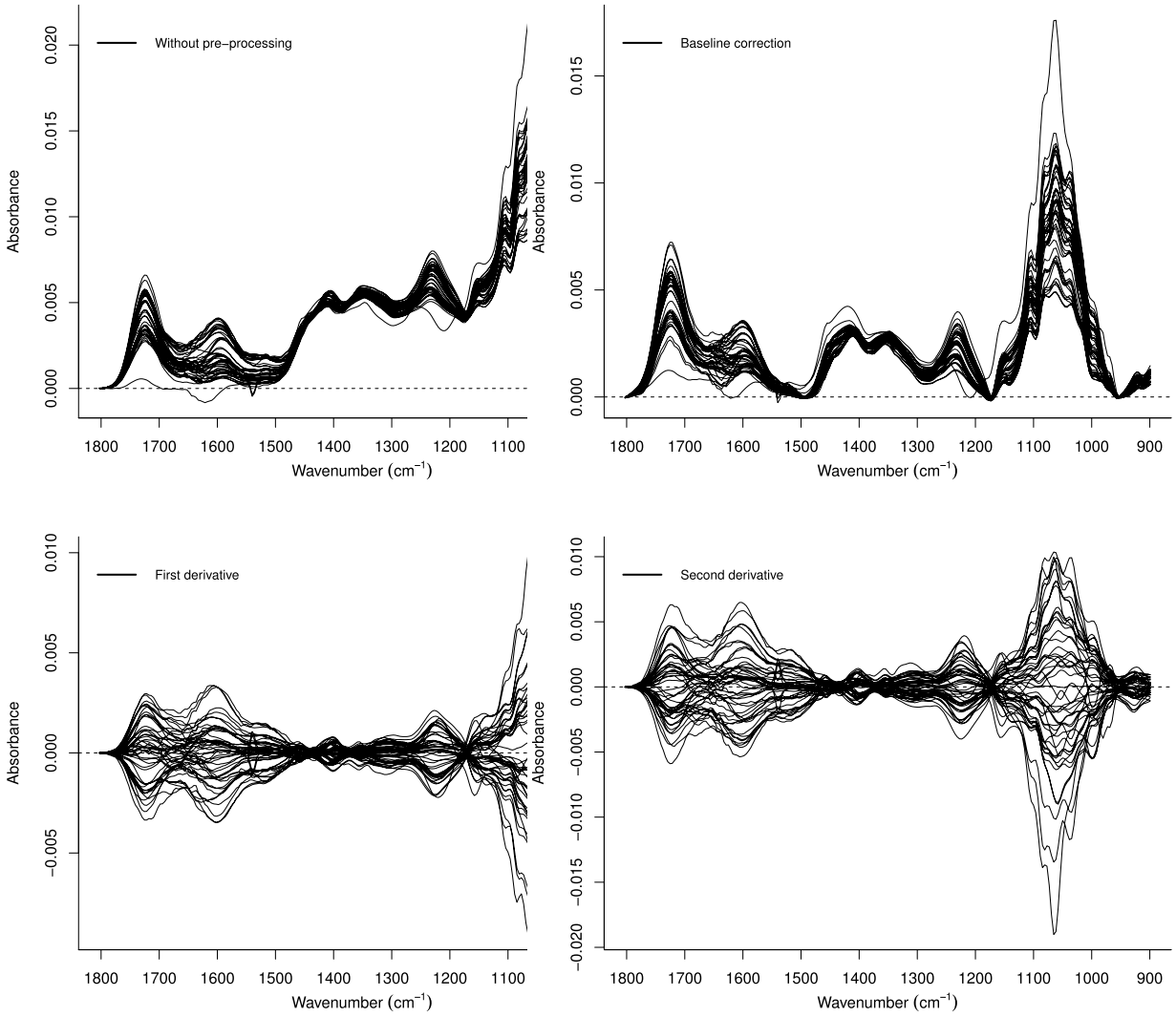

Fig. 3. Spectra of dataset 1 obtained from: without preprocessing, baseline correction preprocessing spectra, first derivative preprocessing spectra, second derivative preprocessing spectra.

Table 4
Information of the datasets used.

Datasets	Training	Testing	Variables	Classes
Dataset 1	632	351	235	2
Dataset 2	80	40	448	3
Dataset 3	36	20	287	2
Dataset 4	80	40	570	4
Dataset 5	89	51	14	–
Dataset 6	38	22	316	–

4.3. Analysis and comparisons for C-PL-ELM algorithm

In this subsection, the performance of the proposed C-PL-ELM algorithm is evaluated and compared using four classification problems and two regression problems. The datasets were randomly divided to generate the training set and the testing set, respectively. We divided all the datasets in approximately 70% for training and 30% for testing to avoid underfitting. The basic information of the datasets are described in Table 4.

For C-PL-ELM algorithm we used two activation functions,

$$\phi(\langle \mathbf{m}, \mathbf{x} \rangle) = \frac{1}{1 + \exp[-(\mathbf{m} \cdot x)]}, \tag{34}$$

and

$$g(\langle \boldsymbol{\varpi}, \mathbf{x} \rangle + b) = \frac{1}{1 + \exp[-(\boldsymbol{\varpi} \cdot \mathbf{x} + b)]}. \tag{35}$$

All the hidden-node parameters $(\mathbf{m}_j, \boldsymbol{\varpi}_j, b_j)_{j=1}^{L}$ are randomly generated based on the uniform distribution.

The user-specified parameters are (C, L), where C is chosen from the range $\{10^{-9}, 10^{-6}, \ldots, 10^{4}, 10^{9}\}$ for the dataset 1 and $\{2^{-20}, 2^{-10}, \ldots, 2^{10}, 2^{20}\}$ for the other datasets. The experiment setup is in accordance to Zheng et al. (2014), that performs a grid search, within a range, to explore different values of C.

The number of hidden nodes L was also determined by grid search using the following ranges: from 60 to 300 for dataset 1, 20 to 200 for dataset 2, 20 to 300 for dataset 3, 50 to 400 for dataset 4, 20 to 300 for dataset 5, and 100 to 500 for dataset 6.

Seen from Fig. 4, C-PL-ELM can achieve a good generalization performance as long as the number of hidden nodes L is large enough. However, our proposal is sensitive to the value of C. In all our simulations on C-PL-ELM with parallel architecture we observed the same behavior. The best parameters found from Fig. 4 are summarized in Table 5.

4.3.1. Activation functions

In this section, we give a detailed analysis of the different activation functions. We combined five activation functions (Sig/Sig, Sig/ Sin,

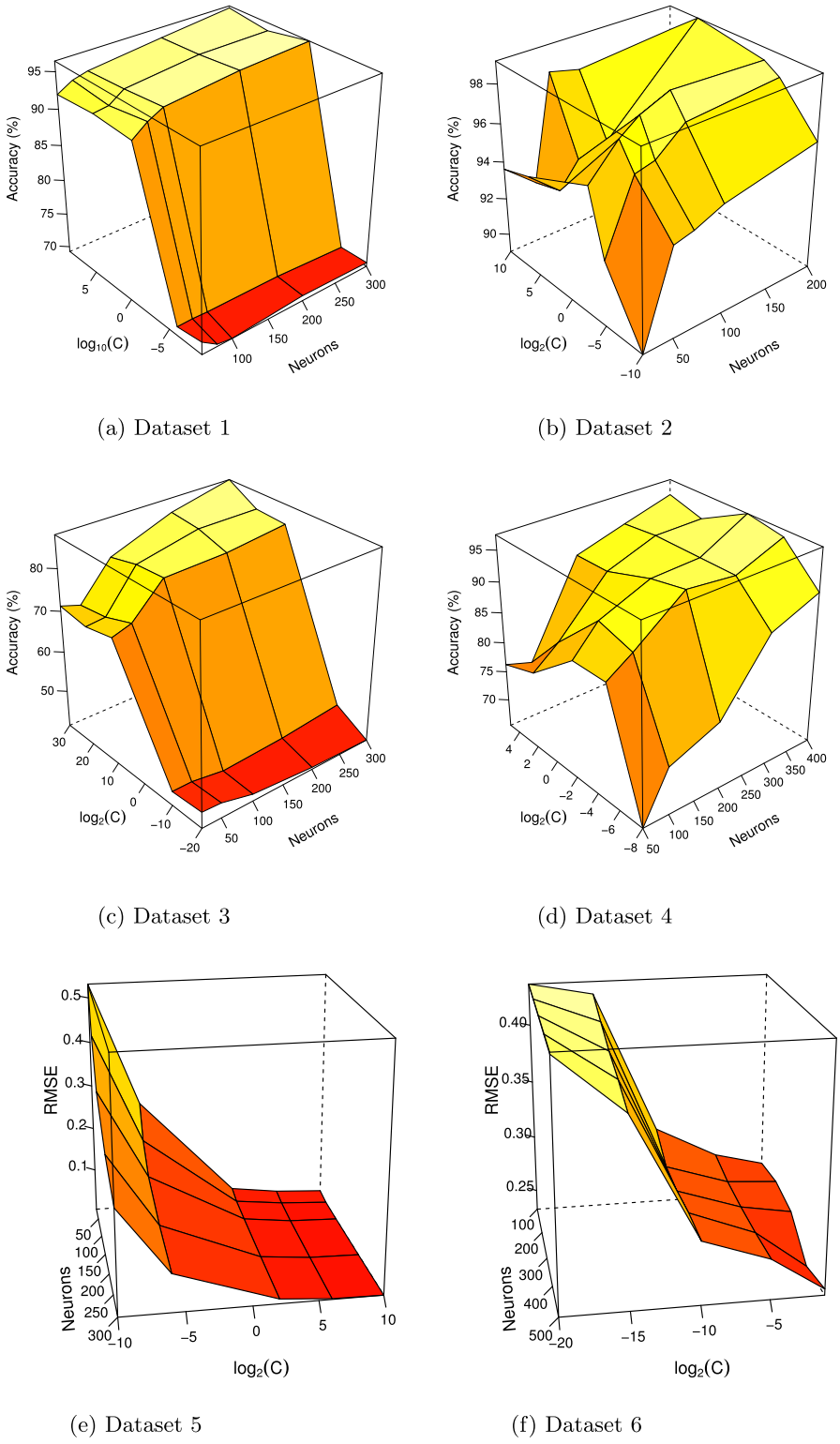

Fig. 4. Performance of C-PL-ELM with a parallel architecture in function of the user-specified parameters (C, L).

Sin/Sin, Hardlim/Sin and Hardlim/Sig). The parameters (C, L) used are those found in Table 5. The results presented in Table 6 show that the combination of activation functions impact on the performance of the proposed algorithm. Overall, we can see that the Sig/Sig activation function always achieves the best performance in all the cases. For classification problems, Sig/Sin and Hardlim/Sin activation functions obtained the worst performances. In terms of RMSE, the worst performance was with Hardlim/Sin (average RMSE of 0.1491 ± 0.0678).

4.3.2. Results

After finding good parameters and the activation function for each layer, we compared the proposed method with ELM (Huang et al., 2006b), C-ELM (Huang et al., 2012) and CO-ELM-R (Wong et al., 2016). In addition, for classification problems, we compared the proposed method with two popular algorithms used for this type of datasets: partial least-squares discriminant analysis (PLS-DA) (Bevilacqua et al., 2012) and support vector machine (SVM) (Li et al., 2009). In regression

Table 5
Best parameters (C, L) found for the datasets.

Datasets	C	L
Dataset 1	10^{-1}	300
Dataset 2	2^{6}	200
Dataset 3	2^{30}	300
Dataset 4	2^{-2}	400
Dataset 5	2^{10}	200
Dataset 6	2^{-1}	400

problems, we compared the proposed method with partial least-squares (PLS) (Wilcox et al., 2016) and support vector regression (SVR) (Bian et al., 2016). For SVM and SVR algorithms, a radial basis function (RBF) was chosen, and the parameters (γ, σ) were selected by grid search. For PLS and PLS-DA models, the number of latent variables (LV) was determined by cross-validation.

For each case, the training set and testing set are randomly generated from the complete dataset before each trial of simulation.

Table 7 shows the results for the four classification problems, where the accuracy (for each model on each dataset) represents the mean and standard deviation of accuracies for all the datasets used. For the four datasets the best performance was obtained with the algorithm C-PL-ELM. The performance is less effective for dataset 3 because it contains too much noise. The best performance non-ELM algorithm for classification problems was the PLS-DA algorithm.

Table 8 shows the average results of 20 trials of simulations for all of these six methods for the two regression problems (dataset 5 and dataset 6). Our proposed method obtained the lowest RMSE values for the two datasets. For non-ELM algorithms, SVR obtained the best performance in terms of RMSE.

In Table 9 we compare our model with the results presented in Zheng et al. (2014). Our results have no pre-processing technique but the authors in Zheng et al. (2014) use the Savitzky-Golay (Savitzky and Golay, 1964) first-order derivative with a 5-point moving window (fitted by a polynomial of two degree). It can be observed in Table 9 that the proposed C-PL-ELM was better in three out of the four datasets when evaluated in the test set. The comparisons of the results with the ones reported in Zheng et al. (2014), can only give a rough idea of the performance, since comparisons are not based on the same partitions and proportions of the training and testing examples.

We statistically analyzed the experimental results with a paired t-test with a confidence level 0.05. From the p-values in Table 10, we can confirm statistically that the proposed algorithm C-PL-ELM outperforms ELM, C-ELM, CO-ELM-R, PLS-DA, SVM, PLS, and SVR.

5. Discussions

The selection of an appropriate pre-processing technique for NIR data is a difficult problem. As mentioned in Rinnan et al. (2009) this can affect the robustness of the model. In the previous sections, we have proposed a novel method to include the noise from the data in the model. We use two Lagrange multipliers as optimization constraints (similar to the concept SVR proposed by Drucker et al. (1997)). We propose an algorithm without using pre-processing techniques. We believe that a robust model must include the noise in near-infrared spectroscopy and should avoid the use of pre-processing techniques.

This paper is a first approach for these types of datasets, we demonstrated that the performance of the proposed algorithm is strongly influenced by the parameters (C, L). As mentioned above, the activation function, the number of hidden nodes and cost parameter are three important parameters requiring to be optimized, which have great effects on the predictive accuracy and stability of proposed model. The best parameters found in this work for the different datasets are shown in Fig. 4. In addition the best performances (on average) were found for the following combinations of activation functions:

$$\text{Sig/Sig} > \text{Hardlim/Sig} > \text{Hardlim/sin} > \text{Sin/Sin} > \text{Sig/Sin}, \quad (36)$$

in classification problems and

$$\text{Sig/Sig} > \text{Sig/SiN} > \text{Hardlim/sig} > \text{Sin/Sin} > \text{Hardlim/Sin}, \quad (37)$$

in regression problems. Where “>" means that the method on the left performs better than the method on the right

In all the previous simulations, we used two hidden layers in parallel as in Henríquez and Ruz (2017b). However, the effect of using more parallel layers has not been studied previously. In Table 11, we show the performance of our proposal using two hidden layers, three hidden layers, four hidden layers, and five hidden layers in parallel. For this simulation, we can see that with two hidden layers in parallel the best performance was obtained.

6. Conclusion

In this paper, we propose a novel algorithm for the analysis in near-infrared spectroscopy. We use the algorithm C-PL-ELM with an architecture in parallel based on a non-linear layer in parallel by another non-linear layer. We incorporate two Lagrange multipliers as optimization constraints with the aim of avoiding the pre-processing of the spectra. The experimental results of this paper are promising

Table 6
Performance of the proposed model for different combination of activation functions.

Datasets	Activation functions				
	Sig/Sig	Sig/Sin	Sin/Sin	Hardlim/Sin	Hardlim/Sig
	Classification problems				
Dataset 1	96.39 ± 0.35	62.27 ± 0.43	54.69 ± 4.30	61.82 ± 2.86	94.82 ± 0.98
Dataset 2	99.13 ± 1.22	58.25 ± 9.77	68.25 ± 9.77	56.25 ± 12.65	71.51 ± 5.41
Dataset 3	87.65 ± 2.04	52.01 ± 10.68	50.75 ± 10.42	52.75 ± 11.29	81.75 ± 4.66
Dataset 4	98.18 ± 1.49	23.25 ± 4.59	23.75 ± 5.16	24.75 ± 5.01	90.11 ± 3.53
Average	95.33 ± 1.28	48.95 ± 6.37	49.36 ± 7.41	48.89 ± 7.95	84.55 ± 3.65
	Regression problems				
Dataset 5	0.0032 ± 0.0003	0.0046 ± 0.0042	0.0072 ± 0.0180	0.0238 ± 0.1072	0.0172 ± 0.0716
Dataset 6	0.2322 ± 0.0172	0.2701 ± 0.0308	0.2848 ± 0.0185	0.2744 ± 0.0285	0.2672 ± 0.0261
Average	0.1177 ± 0.0008	0.1373 ± 0.0175	0.1460 ± 0.0183	0.1491 ± 0.0678	0.1422 ± 0.0488

Table 7
Results for classification problems in terms of accuracy (%).

Datasets	ELM	C-ELM	CO-ELM-R	C-PL-ELM	PLS-DA	SVM
Dataset 1	95.34 ± 0.95	96.11 ± 0.50	96.12 ± 0.48	96.39 ± 0.35	95.12 ± 2.30	94.87 ± 1.87
Dataset 2	97.12 ± 2.72	96.37 ± 2.97	98.25 ± 1.16	99.13 ± 1.22	97.45 ± 1.56	96.40 ± 2.21
Dataset 3	84.75 ± 3.43	86.01 ± 3.01	86.15 ± 2.25	87.65 ± 2.04	79.44 ± 2.17	55.10 ± 2.98
Dataset 4	97.12 ± 2.12	97.13 ± 1.59	97.09 ± 1.89	98.18 ± 1.49	96.67 ± 1.97	95.78 ± 1.77

Table 8
Results for regression problems in terms of RMSE.

	ELM	C-ELM	CO-ELM-R	C-PL-ELM	PLS	SVR
Dataset 5	0.0096 ± 0.0042	0.0036 ± 0.0008	0.0035 ± 0.0005	0.0032 ± 0.0003	0.0264 ± 0.0078	0.0109 ± 0.0096
Dataset 6	0.2421 ± 0.0116	0.2602 ± 0.0159	0.2458 ± 0.0151	0.2322 ± 0.0172	0.4233 ± 0.0091	0.3469 ± 0.0198

Table 9
Comparison between (Zheng et al., 2014) and C-PL-ELM in terms of accuracy (%).

Datasets	Zheng et al. (2014)		Proposed method
	LS-SVM	ELM	C-PL-ELM
Dataset 1	96.01 ± 1.88	95.05 ± 3.15	96.39 ± 0.35
Dataset 2	95.83 ± 2.98	97.78 ± 3.41	99.13 ± 1.22
Dataset 3	97.50 ± 1.73	100.00 ± 0.00	87.65 ± 2.04
Dataset 4	95.08 ± 3.92	97.35 ± 3.65	98.18 ± 1.49

Table 10
Statistical significance test for different simulations. The ✓mark indicates that these two methods are statistically significantly different.

Datasets	M vs M1	M vs M2	M vs M3	M vs M4	M vs M5
Dataset 1	✓	✓	✓	✓	✓
Dataset 2	✓	✓	✓	✓	✓
Dataset 3	✓	✓	✓	✓	✓
Dataset 4		✓	✓	✓	✓
Dataset 5	✓	✓	✓	✓	✓
Dataset 6	✓	✓	✓	✓	✓

M is the proposed C-PL-ELM algorithm.
M1 is the ELM algorithm (Huang et al., 2006b).
M2 is the C-ELM algorithm (Huang et al., 2012).
M3 is the CO-ELM-R algorithm (Wong et al., 2016).
M4 is the PLS-DA or PLS algorithm.
M5 is the SVM or SVR algorithm.

Table 11
Performance of our proposal (C-PL-ELM) with different number of hidden layers in parallel using dataset 1.

Number of hidden layers in parallel	Accuracy (%)
two	96.39 ± 0.35
three	95.45 ± 0.65
four	95.14 ± 0.86
five	93.66 ± 1.09

and indicate that C-PL-ELM has a good performance in the presence of spectra with noise. More research in this direction should be considered with robust models for regression and classification problems for NIR data with noise.

Acknowledgment

The authors would like to thank CONICYT-Chile under grant CONICYT Doctoral scholarship (2015-21150790) (P.H.), Basal(CONICYT)-CMM (G.A.R), and the Research Center Millennium Nucleus Models of Crisis, Chile (NS130017) (G.A.R), for financially supporting this research.

References

Al-Jowder, O., Kemsley, E., Wilson, R., 1997. Mid-infrared spectroscopy and authenticity problems in selected meats: a feasibility study. Food Chem. 59 (2), 195–201.
Alamar, P.D., Caramês, E.T., Poppi, R.J., Pallone, J.A., 2016. Quality evaluation of frozen guava and yellow passion fruit pulps by NIR spectroscopy and chemometrics. Food Res. Int. 85, 209–214.
Alamprese, C., Amigo, J.M., Casiraghi, E., Engelsen, S.B., 2016. Identification and quantification of turkey meat adulteration in fresh, frozen-thawed and cooked minced beef by ft-nir spectroscopy and chemometrics. Meat Sci. 121, 175–181.
Bennett, K.P., Mangasarian, O.L., 1992. Robust linear programming discrimination of two linearly inseparable sets. Optim. Methods Softw. 1 (1), 23–34.
Bevilacqua, M., Bucci, R., Magrì, A.D., Magrì, A.L., Marini, F., 2012. Tracing the origin of extra virgin olive oils by infrared spectroscopy and chemometrics: A case study. Anal. Chim. Acta 717, 39–51.
Bevilacqua, M., Bucci, R., Materazzi, S., Marini, F., 2013. Application of near infrared (NIR) spectroscopy coupled to chemometrics for dried egg-pasta characterization and egg content quantification. Food Chem. 140 (4), 726–734, Special Issue: Food Quality Evaluation.
Bian, X.H., Li, S.J., Fan, M.R., Guo, Y.G., Chang, N., Wang, J.J., 2016. Spectral quantitative analysis of complex samples based on the extreme learning machine. Anal. Methods 8, 4674–4679.
Bian, X., Zhang, C., Tan, X.Y., Dymek, M., Guo, Y., Lin, L., Cheng, B., Hu, X., 2017. Boosting extreme learning machine for near-infrared spectral quantitative analysis of diesel fuel and edible blend oil samples. Anal. Methods.
Briandet, R., Kemsley, E.K., Wilson, R.H., 1996. Discrimination of arabica and robusta in instant coffee by fourier transform infrared spectroscopy and chemometrics. J. Agricult. Food Chem. 44 (1), 170–174.
Cao, J., Zhang, K., Luo, M., Yin, C., Lai, X., 2016. Extreme learning machine and adaptive sparse representation for image classification. Neural Netw. 81, 91–102.
Chen, Q., Ding, J., Cai, J., Zhao, J., 2012. Rapid measurement of total acid content (TAC) in vinegar using near infrared spectroscopy based on efficient variables selection algorithm and nonlinear regression tools. Food Chem. 135 (2), 590–595.
Chen, J., Zhu, S., Zhao, G., 2017. Rapid determination of total protein and wet gluten in commercial wheat flour using sisvr-nir. Food Chem. 221, 1939–1946.
Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.
Despagne, F., Luc Massart, D., 1998. Neural networks in multivariate calibration. Analyst 123, 157R–178R.
Ding, X., Ni, Y., Kokot, S., 2015. NIR spectroscopy and chemometrics for the discrimination of pure, powdered, purple sweet potatoes and their samples adulterated with the white sweet potato flour. Chemometr. Intell. Lab. Syst. 144, 17–23.
Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V., 1997. Support vector regression machines. In: Advances in Neural Information Processing Systems, Vol. 9. MIT Press, pp. 155–161.
Fearn, T., 2008. The interaction between standard normal variate and derivatives. NIR News 19, 16–17.
Feo, J.C., Castro, M.A., Robles, L.C., Aller, A.J., 2004. Fourier-transform infrared spectroscopic study of the interactions of selenium species with living bacterial cells. Anal. Bioanal. Chem. 378 (6), 1601–1607.
Frizon, C.N., Oliveira, G.A., Perussello, C.A., Peralta-Zamora, P.G., Camlofski, A.M., Rossa, Überson B., Hoffmann-Ribani, R., 2015. Determination of total phenolic compounds in yerba mate (ilex paraguariensis) combining near infrared spectroscopy (NIR) and multivariate analysis. LWT - Food Sci. Technol. 60 (2, Part 1), 795–801.
Gorban, A.N., Tyukin, I.Y., Prokhorov, D.V., Sofeikov, K.I., 2016. Approximation with random bases: Pro et contra. Inform. Sci. 364–365, 129–145.
Hajnayeb, A., Ghasemloonia, A., Khadem, S., Moradi, M., 2011. Application and comparison of an ANN-based feature selection method and the genetic algorithm in gearbox fault diagnosis. Expert Syst. Appl. 38 (8), 10205–10209.
Henríquez, P.A., Ruz, G.A., 2017a. An empirical study of the hidden matrix rank for neural networks with random weights. In: 2017 16th IEEE International Conference on Machine Learning and Applications, ICMLA. pp. 883–888.
Henríquez, P.A., Ruz, G.A., 2017b. Extreme learning machine with a deterministic assignment of hidden weights in two parallel layers. Neurocomputing 226, 109–116.
Henríquez, P.A., Ruz, G.A., 2018a. A non-iterative method for pruning hidden neurons in neural networks with random weights. Appl. Soft Comput. 70, 1109–1121.
Henríquez, P.A., Ruz, G.A., 2018b. Twitter sentiment classification based on deep random vector functional link. In: 2018 International Joint Conference on Neural Networks, IJCNN. pp. 1–6.
Holland, J.K., Kemsley, E.K., Wilson, R.H., 1998. Use of fourier transform infrared spectroscopy and partial least squares regression for the detection of adulteration of strawberry purées. J. Sci. Food Agric. 76 (2), 263–269.
Huang, G.B., Chen, L., Siew, C.K., 2006a. Universal approximation using incremental constructive feedforward networks with random hidden nodes. Trans. Neural Netw. 17 (4), 879–892.
Huang, G.B., Zhou, H., Ding, X., Zhang, R., 2012. Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst. Man Cybern. B 42 (2), 513–529.
Huang, G.B., Zhu, Q.Y., Siew, C.K., 2004. Extreme learning machine: a new learning scheme of feedforward neural networks. In: Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on, vol. 2. pp. 985–990.
Huang, G.B., Zhu, Q.Y., Siew, C.K., 2006b. Extreme learning machine: Theory and applications. Neurocomputing 70 (1–3), 489–501.
Jiang, H., Zhu, W., 2013. Determination of pear internal quality attributes by fourier transform near infrared (FT-NIR) spectroscopy and multivariate analysis. Food Anal. Methods 6 (2), 569–577.
Keithley, R., Mark Wightman, R., Heien, M., 2009. Multivariate concentration determination using principal component regression with residual analysis. TrAC - Trends Anal. Chem. 28 (9), 1127–1136.

Kim, S.B., Temiyasathit, C., Bensalah, K., Tuncel, A., Cadeddu, J., Kabbani, W., Mathker, A.V., Liu, H., 2010. An effective classification procedure for diagnosis of prostate cancer in near infrared spectra. Expert Syst. Appl. 37 (5), 3863–3869.

Li, H., Liang, Y., Xu, Q., 2009. Support vector machines and its applications in chemistry. Chemometr. Intell. Lab. Syst. 95 (2), 188–198.

Li, M., Wang, D., 2017. Insights into randomized algorithms for neural networks: Practical issues and common pitfalls. Inform. Sci. 382–383, 170–178.

Li, X., Zhang, Y., He, Y., 2016. Rapid detection of talcum powder in tea using FT-IR spectroscopy coupled with chemometrics. Sci. Rep. 6, 30313 EP –.

Liu, Y., Pan, X., Wang, C., Li, Y., Shi, R., 2015b. Predicting soil salinity with vis–nir spectra after removing the effects of soil moisture using external parameter orthogonalization. PLoS One 10 (10), 1–13.

Liu, C., Yang, S.X., Deng, L., 2015a. A comparative study for least angle regression on nir spectra analysis to determine internal qualities of navel oranges. Expert Syst. Appl. 42 (22), 8497–8503.

Lorente, D., Escandell-Montero, P., Cubero, S., Gómez-Sanchis, J., Blasco, J., 2015. Visible–nir reflectance spectroscopy and manifold learning methods applied to the detection of fungal infections on citrus fruit. J. Food Eng. 163, 17–24.

Luypaert, J., Massart, D., Heyden, Y.V., 2007. Near-infrared spectroscopy applications in pharmaceutical analysis. Talanta 72 (3), 865–883.

Mabood, F., Gilani, S.A., Albroumi, M., Alameri, S., Nabhani, M.M.A., Jabeen, F., Hussain, J., Al-Harrasi, A., Boqué, R., Farooq, S., Hamaed, A.M., Naureen, Z., Khan, A., Hussain, Z., 2017. Detection and estimation of super premium 95 gasoline adulteration with premium 91 gasoline using new NIR spectroscopy combined with multivariate methods. Fuel 197, 388–396.

Ouyang, Q., Chen, Q., Zhao, J., Lin, H., 2013. Determination of amino acid nitrogen in soy sauce using near infrared spectroscopy combined with characteristic variables selection and extreme learning machine. Food Bioprocess Technol. 6 (9), 2486–2493.

del P. Castillo, R., Araya, J., Troncoso, E., Vinet, S., Freer, J., 2015. Fourier transform infrared imaging and microscopy studies of pinus radiata pulps regarding the simultaneous saccharification and fermentation process. Anal. Chim. Acta 866, 10–20.

Pao, Y.H., Park, G.H., Sobajic, D.J., 1994. Learning and generalization characteristics of the random vector functional-link net. Neurocomputing 6 (2), 163–180.

Park, J.I., Liu, L., Ye, X.P., Jeong, M.K., Jeong, Y.S., 2012. Improved prediction of biomass composition for switchgrass using reproducing kernel methods with wavelet compressed ft-nir spectra. Expert Syst. Appl. 39 (1), 1555–1564.

Peng, J., Li, L., Tang, Y.Y., 2013. Combination of activation functions in extreme learning machines for multivariate calibration. Chemometr. Intell. Lab. Syst. 120, 53–58.

Pierna, J.F., Lecler, B., Conzen, J., Niemoeller, A., Baeten, V., Dardenne, P., 2011. Comparison of various chemometric approaches for large near infrared spectroscopic data of feed and feed products. Anal. Chim. Acta 705 (1–2), 30–34, A selection of papers presented at the 12th International Conference on Chemometrics in Analytical Chemistry.

Rinnan, A., van den Berg, F., Engelsen, S.B., 2009. van den Berg F Engelsen S.B. Review of the most common pre-processing techniques for near-infrared spectra. TRAC Trends Anal. Chem. 28 (10), 1201–1222.

Rosipal, R., Trejo, L.J., 2001. Kernel partial least squares regression in reproducing kernel hilbert space. J. Mach. Learn. Res. 2 (Dec), 97–123.

Samat, A., Du, P., Liu, S., Li, J., Cheng, L., 2014. E^2LMs : Ensemble extreme learning machines for hyperspectral image classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 7 (4), 1060–1069.

Savitzky, A., Golay, M.J.E., 1964. Smoothing and differentiation of data by simplified least squares procedures. Anal. Chem. 36 (8), 1627–1639.

Schmidt, W.F., Kraaijveld, M.A., Duin, R.P.W., 1992. Feedforward neural networks with random weights. In: Proceedings. 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems. pp. 1–4..

Smola, A.J., Schölkopf, B., 2004. A tutorial on support vector regression. Stat. Comput. 14 (3), 199–222.

Stevens, A., Ramirez-Lopez, L., 2013. An introduction to the prospectr package. R package version 0.1.3.

Tapp, H.S., Defernez, M., Kemsley, E.K., 2003. FTIR spectroscopy and multivariate analysis can distinguish the geographic origin of extra virgin olive oils. J. Agricult. Food Chem. 51 (21), 6110–6115.

Tyukin, I.Y., Prokhorov, D.V., 2009. Feasibility of random basis function approximators for modeling and control. In: 2009 IEEE Control Applications, (CCA) Intelligent Control, ISIC. pp. 1391–1396.

Wilcox, K.E., Blanch, E.W., Doig, A.J., 2016. Determination of protein secondary structure from infrared spectra using partial least-squares regression. Biochemistry 55 (27), 3794–3802.

Wong, S.Y., Yap, K.S., Yap, H.J., 2016. A constrained optimization based extreme learning machine for noisy data regression. Neurocomputing 171, 1431–1443.

Xu, L., Shi, W., Cai, C.B., Zhong, W., Tu, K., 2015. Rapid and nondestructive detection of multiple adulterants in kudzu starch by near infrared (NIR) spectroscopy and chemometrics. LWT - Food Sci. Technol. 61 (2), 590–595.

Yang, L., Sun, Q., 2016. Comparison of chemometric approaches for near-infrared spectroscopic data. Anal. Methods 8, 1914–1923.

Ye, M., Gao, Z., Li, Z., Yuan, Y., Yue, T., 2016. Rapid detection of volatile compounds in apple wines using ft-nir spectroscopy. Food Chem. 190, 701–708.

Zhang, L., Suganthan, P., 2016a. A comprehensive evaluation of random vector functional link networks. Inform. Sci. 367–368, 1094–1105.

Zhang, L., Suganthan, P., 2016b. A survey of randomized algorithms for training neural networks. Inform. Sci. 364–365, 146–155.

Zheng, W., Fu, X., Ying, Y., 2014. Spectroscopy-based food classification with extreme learning machine. Chemometr. Intell. Lab. Syst. 139, 42–47.