Original article

Learning styles' recognition in e-learning environments with feed-forward neural networks

J. E. Villaverde*, **D. Godoy***† **& A. Amandi***†
*ISISTAN Research Institute, UNICEN University, Campus Universitario, Paraje Arroyo Seco, Tandil (CP 7000), Bs. As., Argentina
†CONICET, Consejo Nacional de Investigaciones Científícas y Técnicas, Av. Rivadavia 1917, Capital Federal (CP1033), Bs. As., Argentina

Abstract People have unique ways of learning, which may greatly affect the learning process and, therefore, its outcome. In order to be effective, e-learning systems should be capable of adapting the content of courses to the individual characteristics of students. In this regard, some educational systems have proposed the use of questionnaires for determining a student learning style; and then adapting their behaviour according to the students' styles. However, the use of questionnaires is shown to be not only a time-consuming investment but also an unreliable method for acquiring learning style characterisations. In this paper, we present an approach to recognize automatically the learning styles of individual students according to the actions that he or she has performed in an e-learning environment. This recognition technique is based upon feed-forward neural networks.

Keywords learning styles, neural networks, Web-based instruction.

Introduction

Students learn in different ways. Some students prefer graphics, such as diagrams and blueprints, while others prefer written material. Some students feel more comfortable with facts, data and experimentation, whereas others prefer principles and theories (Felder & Silvermann 1988). E-learning environments can take advantage of these different forms of learning by recognizing the style of each individual student using the system and adapting the content of courses to match this style.

There are a few systems that are actually capable of adapting courses' contents according to students' learning styles (Carver *et al*. 1999; Gilbert & Han 1999; Paredes & Rodriguez 2002; Stash & Brau 2004). In these systems, the learning materials are then presented in the way that best fit the learning style of each student, which is usually assessed through a predefined questionnaire. However, answering long questionnaires is a time-consuming task that students are not always willing to carry out and, consequently, results become unreliable (Stash *et al*. 2004).

This paper describes an approach to the problem of mapping students&apso; actions within e-learning environments into learning styles. The method is based on artificial neural networks (ANNs). Neural networks are computational models for classification inspired by the neural structure of the brain: models that have proven to produce very accurate classifiers. In the proposed approach, feed-forward neural networks are used to recognize students' learning styles based upon the actions they have performed in an e-learning system.

The rest of this paper is organized as follows: The next section briefly describes learning style theory. This is followed by an overview of ANNs and the back-propagation algorithms. The sections after that

Accepted: 16 March 2006
Correspondence: Jorge E. Villaverde, ISISTAN Research Institute, UNICEN University, Campus Universitario, Paraje Arroyo Seco, Tandil (CP 7000), Bs. As., Argentina. E-mail: jvillave@exa.unicen.edu.ar

show how we have modelled the recognition of learning styles using neural networks and empirical results obtained in the recognition of learning styles. The penultimate section discusses related work and the last section presents our conclusions.

Learning styles

Keefe (1979) defines learning styles as characteristic cognitive, affective and psychological behaviours that serve as relatively stable indicators of how learners perceive, interact with and respond to learning environments. In the last decade, several learning styles models have been proposed (Kolb 1984; Myers & McCaulley 1985; Felder & Silvermann 1988; Lawrence 1993; Litzinger & Osif 1993). A critical review of learning styles, analysing their reliability, validity and implication for pedagogy, can be found in (Coffield *et al*. 2004a;b). The authors of this review concluded that in the field of learning styles, there is a lack of theoretical coherence and a common framework (Coffield *et al*. 2004b, p. 145).

In spite of this fact, experimental research on the application of learning styles in computer-based education provides support for the view that learning can be enhanced through the presentation of materials that are consistent with a student's particular learning style (Budhu 2002; Peña *et al*. 2002; Stash *et al*. 2004). For example, it has been shown that the performance of students in a simple Web-based learning environment correlates with their self-reported learning preference (Walters *et al*. 2000).

In this paper, we have adopted the model suggested by Felder and Silverman (1988) for engineering education, which classifies students according to their position in several scales that evaluate how they perceive and process information.

This model classifies students according to four dimensions:

- *Perception*: What type of information does the student preferentially perceive: *sensory* (external)——sights, sounds, physical sensations, or *intuitive* (internal)——possibilities, insights, hunches?
  - *Sensory learners*: these students like facts, data and experimentation. They perceive concrete, practical, and are oriented towards facts and procedural information. When solving problems, sensory students are routinely very patient with details and usually dislike surprises. Because of these characteristics, they show a slower reaction to problems, but they typically present a better outcome.
  - *Intuitive learners*: intuitive learners prefer theories and principles. They rapidly become bored with details and mechanical problem solving. Innovation is what attracts intuitive learners' attention. They generally solve problems quickly, not paying much attention to details. This makes them fast but prone to errors and, then, they often get lower qualifications than sensitive learners.
- *Input*: Through which sensory channel is external information most effectively perceived: *visual*——pictures, diagrams, graphs, demonstrations or *verbal*——written or spoken sounds?
  - *Visual learners*: they remember, understand and assimilate information better if it is presented to them in a visual way. They tend to remember graphics, pictures, diagrams, time lines, blueprints, presentations and any other visual material.
  - *Verbal learners*: cognitive scientists have established that our brains generally convert written works into their spoken equivalents and process them in the same way that they process spoken words (Felder & Brent 2005). Hence, verbal learners are not only those who prefer auditory material but also those who remember well what they hear and what they read.
- *Processing*: How does the student prefer to process information: *actively*——through engagement in physical activities or discussions, or *reflectively*——through introspection?
  - *Active learners*: they feel more comfortable with active experimentation than with reflexive observation. An active person learns by trying things out and working with others. They like doing something in the external world with the received information. Active learners work well in groups and in situations that require their participation.

- *Reflective learners*: reflective learners prefer introspective examination and manipulation of information. They learn by thinking things through and working alone or with another person.

- *Understanding*: How does the student progress towards understanding: *sequentially*——in continual steps, or *globally*——in large jumps, holistically?

- *Sequential learners*: sequential learners follow a line of reasoning when progressing towards the solution of a problem; they like things to be linear. They learn better if information is presented in a steady progression of complexity and difficulty (i.e. they learn in small incremental steps).

- *Global learners*: global learners make intuitive leaps and may be unable to explain how they come up with solutions. They are holistic, system thinkers; they learn in large leaps. They need to understand the whole before understanding the parts that compose it; they need to get the '*big picture*'.

If the dimensions were absolute, these four dimensions of learning allow to obtain 16 different learning styles (i.e. in the perception dimension, a student would either be sensitive or intuitive). However, each dimension can be rated on a scale, for example the scale used in the Index of Learning Styles (ILS)[1] is a scale of 22 degree. Having this scale, the number of different styles is 234256 (22^4).

In the following section, we detail how ANNs can be used to recognize the learning styles of students based on the actions they perform in an e-learning system.

ANNs

The human brain is composed of cells (neurons) that are the only cells capable of communicating with each other. This is one of the capabilities that allows humans to exhibit intelligent behaviour. ANNs are computational models based on the biological neural structure of the brain, as first proposed by McCulloch and Pitts (1943). This computational model, also known as connectionism, aims to mathematically represent and reproduce the way a human nervous system works.

[1]Available online at http://www.ncsu.edu/felderpublic.ILSpage.html

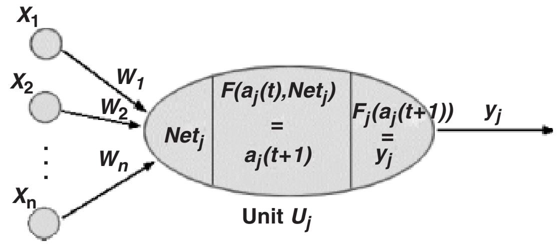

Fig 1 Main components of an artificial neuron.

The neuron is the basic processing unit. Each neuron receives signals from others neurons, processes these signals and transmits an output to its neighbours. Some of the signals it receives excite it while others inhibit it. Usually, an exciting signal is represented as a positive real value, while a negative value is assigned to an inhibiting signal. As shown in Fig 1, the *Net* input of a neuron is the sum of values that arrive at its input. This value represents the excitation level of the neuron. If it exceeds a certain threshold, the neuron fires an output signal to its neighbours. From this input–process–output model, neurons are classified with respect to where the signals come from. If the signals that the neuron receives come from the environment, it is called an *input neuron*. If it receives the input from other neurons and transmits its outcome to others as well, it is called a *hidden neuron*. Finally, if it sends its output to the environment, then it is called an *output neuron*. Usually, neurons sharing similar characteristics are grouped together, forming layers of neurons. What distinguishes one layer of neurons from another are their inputs and outputs. So, neurons belonging to layer i receive their input from layer i-1, and they send their output to layer $i+1$. Three kinds of layers are distinguished in ANN literature. If the layer contains input neurons, it is an *input layer*; if it contains hidden neurons, it is a *hidden layer*; and if it contains output neurons then it is an *output layer*.

An ANN is termed *feed-forward*, if neurons belonging to layer i receive input only from layer i-1, and only send output to layer $i+1$. This means that in feed-forward neural networks, there are neither connection between neurons from the same layer (*lateral connections*) nor from previous layers (*recurrent connections*). Information in this class of networks only flows forward, from the input to the output passing through hidden layers.

Structure and operation of ANNs

The fundamental unit of processing in ANNs is the Artificial Neuron (AN); see Fig 1. ANs are mathematical models that represent the general characteristics of biological neurons. Let us see how these biological features are mathematically modeled and how they contribute to the learning process of ANNs.

When a signal is transmitted from processing unit i to processing unit j, the signal x_i is modified by the synaptic weight (w_{ij}) associated with this communication channel. The modulated signals that arrive at unit j are added to form the net input Net_j as is shown in

$$Net_j = \sum_i x_i w_{ij}, \tag{1}$$

Each neuron is characterized in any instant of time t by a real value, called activation state or activation level, $a_j(t)$ Also, there is a function F, called *activation function*, which determines the next activation state based on the current activation state and the Net_j input of the neuron.

$$F\left(a_j(t), Net_j\right) = a_j(t+1). \tag{2}$$

Associated with each unit there exists an output function, f_j, that transforms the current activation level into an output signal y_i. This signal is sent through a unidirectional communication channel to other units in the network

$$f_j\left(a_j(t+1)\right) = y_j \tag{3}$$

Learning algorithm

A learning algorithm is the process by which an ANN generates internal changes so that it can adapt its behaviour in response to the environment. The modifications that by the network during this process enable it to gain better performance, so it can overcome its output to the environment. When there is an external agent involved in the learning process, it receives the name of *supervised learning*. Back-propagation is a supervised learning algorithm, used in feed-forward neural networks, which reduce the global error produced by the network over the weight space.

Back-propagation (Parker 1982; Werbos 1988) is a generalization of the LMS algorithm (Rosenblatt 1958; Widrow & Hoff 1960) applied to feed-forward multi-layer perceptron networks. This network is one of the most spread models in the connectionism field, due to its learning capability to associate an input space to an output. It has been demonstrated by different authors that a multi-layer perceptron is a universal function approximation mechanism, in the sense that any continuous function over a compact R^n can be approximated by a multi-layer perceptron (Cybenko 1989; Hornik *et al*. 1989).

BPN nets operate in two steps. In the first step, called training process, the network is initialized with random small values in its weights. The goal of this process is to find a set of weight values that minimize the global error of the network, given in equation (5). The second step is called generalization. In this step, the network has already learned an internal representation of the previously presented patterns and becomes able to classify novel patterns presented as inputs.

The learning process of a BPN is briefly described as follows: a pattern is a two-tuple $P_i = \langle X_i, T_i \rangle$ where X_i is a set of values that will be presented as input to the network and T_i is a set of values that represent the desired target for the values presented at the input layer. For the network to learn a pattern, P_i, the values of the pattern have to be presented at the input layer first. These values are taken as stimulus and are propagated forward until the output layer is reached. In the output, a set (O) of values obtained by the network (called o_j) are compared with the set of desired values (T_i) to obtain the pattern error according to

$$Err_i = \sum_{j=1}^{M} \left(o_j - t_j\right)^2, \tag{4}$$

where M is the number of neurons in the output layer, $o_i \in O$ and $t_j \in T$. Hence, the global error of the network is calculated considering all patterns as follows:

$$Err = \frac{1}{2P} \sum_{k=1}^{P} \sum_{j=1}^{M} \left(o_j^{(k)} - t_j^{(k)}\right)^2, \tag{5}$$

where P is the number of patterns in the training set. This error is the one that the training procedure tries to minimize. These error values are back propagated through the network to adjust the values of its connection weights so that the change in values is proportional to the gradient descent of the error in equation (6). This weight adjustment rule is known as the generalized delta rule (GDR) (Rumelhart *et al*.

1986;1988) and is a generalization of the Widrow–Hoff delta rule (Widrow & Hoff 1960)

$$\begin{aligned}\Delta W_{ij} &= \frac{\partial Err}{\partial W_{ij}} \\ &= \frac{\partial}{\partial W_{ij}} \frac{1}{2} \sum_{k=1}^{M} (o_k - t_k)^2 \\ &= \sum_{k=1}^{M} (o_k - t_k) \frac{\partial o_k}{\partial W_{ij}}.\end{aligned} \tag{6}$$

This equation can also be rewritten as equation (7), with addition of the momentum (Rumelhart *et al.* 1986) term to achieve faster convergence. The momentum part of the equation, $\beta(W(t-1) - W(t-2))$, allows a network to respond not only to the local gradient but also to recent trends in the error surface.

$$\begin{aligned}W(t) = W(t-1) - \alpha\delta_j y_j \\ + \beta(W(t-1) - W(t-2)),\end{aligned} \tag{7}$$

where δ_j is calculated according to the rule by

$$\delta_j = \begin{cases} f'(z_j)(o_j - t_j) & (output : layer) \\ f'(z_j) \sum\limits_i W_{ij}\delta j & (hidden : layer) \end{cases} \tag{8}$$

This process of weight adjustment is repeated until a desired error threshold is reached.

Modelling learning styles with feed-forward neural networks

The neural network architecture that we propose in this paper aims to find a mapping between students' actions in the system and the learning style that they best fit. To achieve this goal, we must identify the inputs of the network, its outputs and the meaning of their possible values. It is also necessary to determine other architectural parameters, such as the number of hidden layers to be used, the number of processing units in each of the hidden layers, the activation function to be used in the processing units and the learning coefficient of the network. Each of these issues is analysed in the following subsections.

Input layer

To represent the input of the network, we propose the use of one processing unit (neuron) in the input layer per observed action in the system. These actions are as follows:

- *Reading material*: academic units can be presented using both abstract (theories) and concrete material (exercises). What kind of material is the student most interested in?
- *Access to examples*: in each academic unit, a number of examples are presented to students. In relation to the total number of available examples, how many of them has the student accessed to?
- *Answer changes*: Does the student change the answers of the exam before he hands it over? If yes, what is the percentage of answers he has changed?
- *Exercises*: a number of exercises are also included in academic units. In relation to the total number of available exercises, how many exercises has the student accessed to?
- *Exam delivery time*: each exam has an associated time –to solve. What is the relation between the student's exam delivery time and the units' time –to solve?
- *Exam revision*: in relation to the time to solve of the exam, what was the percentage of time spent by the student checking the correctness of the exam?
- *Chat usage*: the student may ignore the chat, read other students' messages or read/write messages with others.
- *Forum usage*: the student may ignore the forum, read other students posted messages or post messages in the forum.
- *Mail usage*: the student may use (or not) the e-mail.
- *Information access*: information in academic units is presented following a line of reasoning. How has the student followed that line of reasoning? Lineally, or has he or she visited a random sequence of items?

These values have to be encoded in the real interval $[-5; +5]$ as expected by the neurons in the input layer of the network. This interval was intentionally selected to match the expected domain of the activation function selected for the units of the net as shown in subsection 'Network architecture and parameters'. Table 1 summarizes the input vector, X, representation.

Output layer

The output of the network should approximate the learning style of the students based on the actions

Table 1. Input representation.

X	Action	-5	$+5$
x_0	Reading material	Abstract	Concrete
x_1	Access to examples	Few	Much
x_2	Answer changes	Few	Much
x_3	Exercises	Few	Much
x_4	Exam delivery time	Quick	Slow
x_5	Exam revision	Few	Much
x_6	Chat usage	Don't use	Read and write
x_7	Forum usage	Don't use	Read and post
x_8	Mail usage	Don't use	Use
x_9	Information access	Lineal	Global

Table 2. Output representation.

O	Dimension	-1	$+1$
o_0	Perception	Intuitive	Sensitive
o_1	Processing	Active	Reflective
o_2	Understanding	Sequential	Global

presented at the input layer. In this case, we propose the use of one processing neuron in this layer per learning style dimension used in the model. In this work, only three of the four dimensions of the Felder–Silverman model have been used to model students' learning style. These dimensions are as follows:

- *Perception*: this dimension determines whether the style of the student is intuitive or sensitive.
- *Processing*: this dimension decides whether a student's leaning style better fits active or reflective.
- *Understanding*: this dimension informs whether the student learning style is sequential or global.

Table 2 summarizes the output vector representation.

Hidden layer

In a multi-layer perceptron using continuous nonlinear hidden-layer activation functions, as proposed in this work, one hidden layer with an arbitrarily large number of units suffices for the 'universal approximation property' (Hornik *et al*. 1989; Bishop 1995). However, there is still no theory about how many hidden units are needed to approximate any given function.

Although there are some empirical rules, such as the Baum–Haussler rule (Baum & Haussler 1988), for determining the desirable number of neurons in the hidden layer of a multi-layer feed-forward neural network, we determined this architectural parameter via trial-and-error experimentation. A total of 24 units have been empirically found as appropriate for the task at hand, considering that this layer has to have enough processing units to represent the nonlinear aspects of the model and not too many for making the training process very complex.

Network architecture and parameters

The network is trained using the GDR (Rumelhart *et al*. 1986;1988), a generalization of the Widrow–Hoff delta rule (Widrow & Hoff 1960). The activation function used in the network units is the hyperbolic tangent function shown in

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{9}$$

Many interesting properties of this function make it suitable to represent the activation function of the network. On the one hand, the function domain can be restricted to $[-5; +5]$ where the function reaches more than 99.99% of its range; the different observed aspects are then projected to this range. Another interesting property is that its derivate can be defined in terms of the function's output; this makes calculation easier as is shown in equation (10). A further important property of the hyperbolic tangent function is that its range is $[-1; +1]$; this property can be utilized to represent the different ranges of the learning style dimensions.

$$\begin{aligned} \frac{d}{dx}\tanh(x) &= \frac{d}{dx}\frac{e^x - e^{-x}}{e^x + e^{-x}} \\ &= \frac{d}{dx}\frac{\sinh(x)}{\cosh(x)} \\ &= \frac{\cosh^2(x) - \sinh^2(x)}{\cosh^2(x)} \\ &= 1 - \tanh^2(x) \end{aligned}. \tag{10}$$

This architecture parameter, along with the proper selection of learning rates and momentum coefficients, defines the specific values selected for this work. The learning rate, α, was set to a small value between 0.1 and 0.25 so that the representation acquired can be a faithful one.

Table 3. Proposed Architectural Parameters.

Parameter	Value
Number of input neurons	10
Number of hidden neurons	24
Number of output neurons	3
Activation function	Hyperbolic tangent
Learning rate	0.02
Momentum	0.5

To sum up, the proposed architecture for learning style recognition is a three-layered feed-forward neural network with Batch Gradient Descent with the Momentum learning algorithm. In this architecture, the first layer contains a total of 10 input neurons; the hidden layer contains 24 processing units that are connected to 3 output units in the third layer. Information about the network architecture and other parameters is summarized in Table 3.

Temporal adaptation

To ensure that the student's learning style does not oscillate between different states (and keeps the system changing the contents of the student's courses as a consequence), historical interaction data can be supplied to the classifier to enhance the prediction of the learning style. To accomplish this goal, a parameter N is defined to represent the length of the input sequence to the network, causing the input to the classifier to be a function of the historical data. That is, the input to the system is a finite sequence X where

$$X = x_t, x_{t-1}, \ldots, x_{t-N}. \qquad (11)$$

This sequence of inputs is pre-processed applying the discount return transformation (Sutton 1988) before becoming the real input to the network. Equation (12) shows how the actual input to the ANN will be produced from the data measured at time t and from historical information

$$x_t = \sum_{k=1}^{N} \gamma^{k-1} x_{t+k} \quad 0 \leq \gamma \leq 1. \qquad (12)$$

Another important network parameter is the discount factor, γ, which determines the importance given to past experience. Along with the length of the sequence, the discount factor allows to control the relevance of past experience in the system to the recognition of the learning style.

Experimental results

Estimating the accuracy of classifiers induced by supervised learning algorithms (i.e. classifier's probability of correctly recognizing a randomly selected instance) is important not only to predict its future prediction accuracy but also for choosing a classifier from a given set (model selection) or combining classifiers (Wolpert 1992). In order to evaluate the proposed approach, we generated an artificial dataset for experimentation by simulating the actions of students.

For this task, we considered that each student has a particular learning style denoted by a set of preferred actions and behaves according to it. The resulting dataset built for testing the network consisted of 100 pairs of input–output values. Each of these pairs contained possible actions that students might perform in the system associated with their corresponding learning style dimensions.

To determine the best number of processing units in the hidden layer, we trained the network varying this parameter and estimated the architecture accuracy using k-fold cross-validation with $k = 10$. Each of the 10 folds was composed of a set of 90 training patterns and 10 test patterns. To measure the accuracy of each classifier, we calculated the global error (equation (5)) that it produces when it is stimulated with the test cases. Each of the output dimensions was considered independently in the calculation.

Figure 2 shows the mean accuracy of cross-validation when varying the number of neurons in the network-hidden layer. We can observe in the figure that the best accuracy (69.3%) is reached when the number of hidden units is 24. Figure 3 shows the variance of the classifier when the number of neuron varies. In this figure, the lowest variance (1.53%) is obtained when the number of neurons is 22. Based on these results, we have chosen 24 neurons for achieving the maximum accuracy. For this value, the variance (1.70%) of the classifier is not the minimum but it is still acceptable.

Related work

Most adaptive educational systems providing individualized courses according to learning styles, such

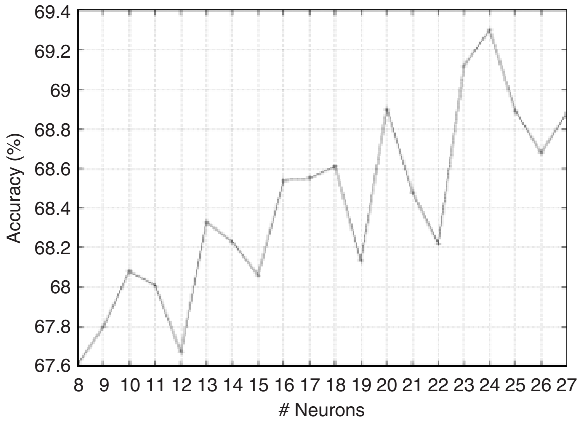

Fig 2 Mean accuracy as a function of the number of neurons in the hidden layer of the network.

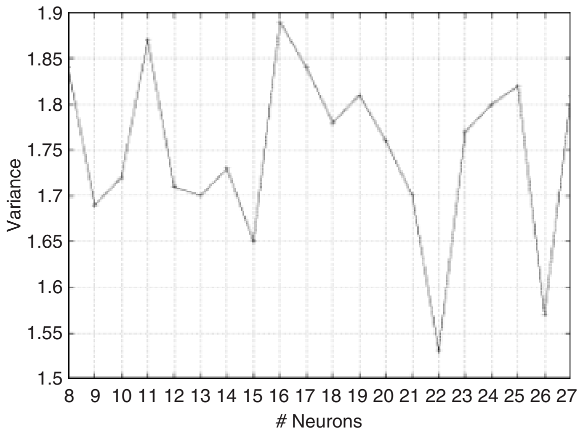

Fig 3 Variance as a function of the number of neurons in the hidden layer of the network.

as ARTHUR (Gilbert & Han 1999), CS388 (Carver *et al*. 1999), MAS-PLANG (Peña *et al*. 2002) and TangoW (Paredes & Rodriguez 2002), assess students' learning styles by having them complete questionnaires. Each question of these questionnaires requires that the student selects one of several possible answers. The main disadvantage of this approach is that the questionnaires tend to be time consuming and students might try to answer questions arbitrarily. This has a negative effect on the accuracy of the results obtained. A further disadvantage is that learning styles can change over time, so the knowledge gathered by the questionnaire can become obsolete.

Conversely, the proposed approach consists in training a neural network to determine learning styles automatically based on the students' actions. In this approach, the network is trained based on a number of the most recent actions performed by students and, thus, it allows the system to deal with changes in the students' learning styles. The closest related work is AHA! (Stash & Brau 2004), an educational system that infers the students' learning style from the observation of their browsing behaviour. In contrast to the mentioned systems, it does not make use of questionnaires. However, this system covers a small number of learning styles, those corresponding to field-dependent/independent styles and to visual/verbal styles, whereas the approach presented in this paper is based on the Felder and Silverman theory for engineering education.

Learning styles in engineering education have been studied by (Al-Holou & Clum 1999; Ayre & Nafalski 2000; Budhu 2002; Kuri & Truzzi 2002). Kuri and Truzzi (2002) use the Felder–Silverman model to assess the distribution of responses of freshmen engineering students using the ILS questionnaire, and they concluded that the use of learning styles to accommodate their teaching strategies to students has been shown to be effective in encouraging students. Other e-learning environments that are not exclusively from engineering education have incorporated the Felder–Silverman model of learning styles (Peña *et al*. 2002; Kim & Moore 2005). They also rely on responses obtained from ILS questionnaires fulfilled at the beginning of the courses, and they adapt the content of the courses to the characteristics of instructional and learning strategies.

Conclusions

In this paper, we have described an approach based on feed-forward neural networks to infer the learning styles of students automatically. We have selected the back-propagation algorithm to train the ANN described in this work. In addition, we have described a neural network architecture that learns the associations between students' actions in e-learning environments and their corresponding Felder–Silvermans learning style of engineering education.

The advantage of this approach is twofold. First, an automatic mechanism for style recognition facilitates the gathering of information about learning preferences, making it imperceptible to students. Second, the proposed algorithm uses the recent history of system usage so that systems using this approach can recognize changes in learning styles or some of their dimensions over time.

The recognition mechanism presented in this paper can be introduced in adaptive e-learning environments to help in the detection of students' learning styles and, thus, conveniently adapt the contents of academic courses that are presented to them. It can also be extended to consider further input actions available in particular e-learning systems or domains.

References

Al-Holou N. & Clum J.A. (1999) Teaching electroscience using computer-based instruction. In *29th ASEE/IEEE Frontiers in Education Conference* (eds D. Budny & G. Bjedov), pp. 24–29. San Juan, Puerto Rico.

Ayre M. & Nafalski A. (2000) Recognising diverse learning styles in teaching and assessment of electronic engineering. In *30th ASEE/IEEE Frontiers in Education Conference* (eds D. Budny & G. Bjedov), pp. 18–23. Kansas City, MO.

Baum E.B. & Haussler D. (1988) What size net gives valid generalization? *Neural Computation* **1**, 151–160.

Bishop C.M. (1995) *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

Budhu M. (2002) Interactive web-based learning using interactive multimedia simulations. In *International Conference on Engineering Education (ICEE 2002)*, pp. 1–6. International Network for Engineering Education & Research, Manchester, UK.

Carver C.A.J., Howard R.A. & Lane W.D. (1999) Enhancing student learning through hypermedia courseware and incorporation of student learning styles. *IEEE Transactions on Education* **42**, 33–38.

Coffield F., Moseley D., Hall E. & Ecclestone K. (2004a) Learning styles and pedagogy in post-16 learning: a systematic and critical review. Technical Report 041543, Learning and Skills Research Centre.

Coffield F., Moseley D., Hall E. & Ecclestone K. (2004b) Should we be using learning styles? What research has to say to practice. Technical Report 041540, Learning and Skills Research Centre.

Cybenko G. (1989) Approximations by superposition of sigmoidal functions. *Mathematics of Control, Signals, and Systems* **2**, 303–314.

Felder R.M. & Brent R. (2005) Understanding student differences. *Journal of Engineering Education* **94**, 57–72.

Felder R.M. & Silvermann L.K. (1988) Learning and teaching styles in engineering education. *Journal of Engineering Education* **78**, 674–681.

Gilbert J.E. & Han C.Y. (1999) Arthur: adapting instruction to accommodate learning style. In *World Conference on the WWW and Internet (WebNet 99)* (eds P.D. Bra & J.J. Leggett), pp. 433–438. Honolulu, Hawaii, USA.

Hornik K., Stinchcombe M. & White H. (1989) Multilayer feedforward networks are universal approximators. *Neural Networks* **2**, 359–366.

Keefe L.W. (1979) Learning style: an overview. In *NASSP's Student learning styles: Diagnosing and prescribing programs*, pp. 1–17.

Kim K.S. & Moore J.L. (2005) Web-based learning: factors affecting students' satisfaction and learning experience. *First Monday* **10**. Available from http://www.firstmonday.dk/issue/issue10-11/kim/index.html

Kolb D.A. (1984) *Experiential Learning: Experience as the Source of Learning and Development*. Prentice-Hall, New York.

Kuri N.P. & Truzzi O.M.S. (2002) Learning styles of freshmen engineering students. In *International Conference on Engineering Education (ICEE 2002)*, Manchester, UK.

Lawrence G. (1993) *People Types and Tiger Stripes: A Practical Guide to Learning Styles*. Center for Applications of Psychological Type. Gainesville, FL.

Litzinger M.E. & Osif B. (1993) Accommodating diverse learning styles: designing instruction for electronic information sources. In *What is Good Instruction Now? Library Instruction for the 90s* (ed. L. Shirato). Pierian Press.

McCulloch W.S. & Pitts W. (1943) A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* **5**, 115–133.

Myers I. & McCaulley M. (1985) *Manual: A Guide to the Development and Use of the Myers–Briggs Type Indicator*. Consulting Psychologists Press., Palo Alto, CA.

Paredes P. & Rodriguez P. (2002) Considering learning styles in adaptive web-based system. In *Proceeding of the 6th World Multiconference on Systemics, Cybernetics and informatics* (eds N. Callaos, M. Margenstern & B. Sanchez), pp. 481–485. Orlardo, FL.

Parker D.B. (1982) Learning logic, invention report. Technical Report S81–64, File 1, Office of Technology Licencing, Stanford University.

Peña C.I., Marzo J.L. & de la Rosa J.L (2002) Intelligent agents in a teaching and learning environment on the Web. In *ICALT2002*. IEEE, Kazan, Russia.

Rosenblatt F. (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review* **65**, 386–408.

Rumelhart D.E., Hinton G.E. & Williams R.J. (1986) Learning representations by back-propagating errors. *Nature* **323**, 533–536.

Rumelhart D., Hinton G. & Williams R. (1988) Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (eds D. Rumelhart & G. McClelland), pp. 318–362. MIT Press, Cambridge, MA.

Stash N. & Brau P.D. (2004) Incorporating cognitive styles in AHA! (The adaptive hypermedia architecture). In *IASTED International Conference* (ed. V. Uskov), pp. 378–383. ACTA Press, Innsbruck, Austria.

Stash N., Cristea A. & Brau P.D. (2004) Authoring of learning styles in adaptive hypermedia: problems and solutions. In *13th International Conference on World Wide Web – Alternate Track Papers & Posters*, pp. 114–123.

Sutton R.S. (1988) Learning to predict by the methods of temporal differences. *Machine Learning* **3**, 9–44.

Walters D., Egert C. & Cuddihy E. (2000) Learning styles and web-based education: a quantitative approach. In *Proceedings from the 9th Annual FACT Conference on Instructional Technology*, pp. 115–117. Malo, Buffalo, NY.

Werbos P.J. (1988) Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks* **1**, 339–356.

Widrow B. & Hoff M.E. (1960) Adaptive switching circuits. In *IRE WESCON Convention Record*, pp. 96–104, New York, NY.

Wolpert D.H. (1992) Stacked generalization. *Neural Networks* **5**, 241–259.