Applications of machine learning techniques to a sensor-network-based prosthesis training system

Chenn-Jung Huang [a,*], Yu-Wu Wang [a], Tz-Hau Huang [a], Chin-Fa Lin [a], Ching-Yu Li [b], Heng-Ming Chen [b], Po Chiang Chen [a], Jia-Jian Liao [c]

[a] Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan
[b] Institute of Electronics Engineering, National Dong Hwa University, Hualien, Taiwan
[c] Institute of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

ARTICLE INFO

ABSTRACT

In the past, the utilization of the limb prosthesis has improved the daily life of amputees or patients with movement disorders. However, a leg-amputee has to take a series of training after wearing a limb prosthesis, and the training results determine whether a patient can use the limb prosthesis correctly in her/his daily life. Limb prosthesis vendors thus desire to offer the leg-amputee a complete and well-organized training process, but they often fail to do so owing to the factors such as the limited support of human resource and financial condition of the amputee. This work proposes a prosthesis training system that the amputees can borrow or buy from the limb prosthesis vendors and train themselves at home. Instant feedback messages provided by the prosthesis training system are used to correct their walking postures during the self-training process. An embedded chip is used as a core to establish a body area sensor network for the prosthesis training system. RFID readers and tags are employed to acquire the 3D positioning information of the amputee's limbs in this work to assist in diagnosing the amputee's walking problem. A series of simulations were conducted and the simulation results exhibit the effectiveness and practicability of the proposed prosthesis training system.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

The leg-amputees who wear lower-limb prostheses suffer from attenuated tactile knowledge of their extension, owing to the natural sensory loss resulting from amputation [1]. Many clinical studies report that amputees exhibit an asymmetrical gait pattern [2–4]. For example, unilateral transtibial amputees generally have higher hip extension, knee flexion, and ankle dorsiflexion in amputated side than those in the unaffected side. Leg-amputees are accustomed to the asymmetrical gait pattern if they do not take a series of training after wearing limb prosthesis. Apart from discomfort, the amputees might suffer from some spinal deformities or muscle injuries. Owing to lacking of exercise, muscular flaccidity might occur and the stumps will be getting swollen resulting in that patients cannot swing their thighs backward, and the shank and the knee joint cannot be straightened. Therefore, an expert had better accompany the amputee during the training process to rectify the amputee's gait and the training expense is accordingly significantly increased.

It was reported in [5] that a microprocessor-controlled prosthetic knee can use onboard electronic sensors to collect real-time data and control stance and swing phase movements as well. The human–machine interface for the prosthesis socket has to be properly designed to achieve satisfactory load transmission, stability, and efficient control for mobility [6–8]. The powered ankle–foot prosthesis is capable of providing human-like ankle work and power during stance and decreasing the metabolic cost of transport compared with the conventional passive-elastic prosthesis [9]. The prosthetic legs outfitted with Bluetooth technology were designed in [10] to make two limbs coordinate each other to control the leg movement. Despite a multitude of technical advances in lower-limb prostheses have been observed in the literature, no existing approaches are able to warn the amputee for the detection of the asymmetrical gait. Though Bluetooth prosthesis [10] can automatically control the leg movement for the amputee, the amputee might suffer from uncomfortable and uncoordinated walking. A well-designed gait training process for rehabilitation is thus still demanded. In the recent positioning technology, Want et al. [11] uses diffuse infrared to develop an active badge positioning system. Users wear badges that transmit infrared signals to provide information about their location to a centralized location service via a network of sensors. However, the infrared signals have the

weakness of moving in a straight line, short transmission distance, and easily disturbed by the environment.

It was observed that some sequelae may arise in the traditional way of gait training due to insufficient gait experts' training or relatively high fee. Moreover, the evaluation of gait training effectiveness relies upon accumulated experiences that gait trainer owns and little explicit criterion can be followed. Accordingly, many general statements are used in training records that cause difficult tracing of therapy process. To tackle above-mentioned problems, several important factors are considered in this work, including setting up the quantitative index for leg-amputees training effectiveness assessment, providing real-time guidance for self-training at home, establishing the mechanism of supervising and tracing training process of leg-amputees, offering easy installation, operation, and maintenance to leg-amputees, and considering the affordability of mid and low income families.

A body area sensor network (BASN) is employed in the work. A BASN consists of multiple interconnected nodes on, near, or within a human body, which together provide sensing, processing, and communication capabilities. Through intelligent computing techniques, a BASN can be enabled to monitor environments, systems, and complex interactions in a range of applications, such as healthcare, fitness, and entertainment. A radio frequency identification (RFID) system is adopted to locate the moving track of patient wearing limb prosthesis, and outputs of the RFID are the relative angular magnitude, location, and elapsed time. The received signal strength indication (RSSI) technique [12] is adopted in this work to compute the transmission distance of RFID based on the information received from RFID. RSSI is a circuit that measures the strength of an incoming radio frequency (RF) signal and generates an output that is equivalent to the signal strength. The ability of the receiver to pick the weakest of signals is referred to as receiver sensitivity. The higher the receiver sensitivity is, the better the link quality is. The circuits are designed to measure the signal strength based on the output voltage. The output voltage is higher if the signal strength is good, and vice versa.

An RFID system consists of RFID tags also known as transponders and readers also called interrogators. Coupled with wired or wireless communications, the applications of RFID technology keep increasing in recent years and RFID systems offer many new possibilities and have given rise to new fields aimed at integrating distributed intelligence software applications into our daily lives. Examples of the RFID applications include toll road systems, passports, automobile immobilizer systems, supply-chain management, and animal identification. A so-called location identification based on dynamic active RFID calibration (LANDMARC) was proposed in [13], and the proposed system uses four readers and sixteen active tags to calculate the position of the tracking tag. The advantage of this system is to utilize the concept of the reference tag, and to improve the overall accuracy of positioning. RFID is suitable for short-range wireless transmission because of low-cost, small size, and easy establishment, and it is suitable for indoor positioning system because of having two-way communication function. RFID systems have also been applied increasingly in telemedicine and healthcare facilities to provide medical information and services. The technology offers great potential benefits and it is the next big step to identify and track patients, objects, and assets, and speed up or eliminate many manual operations in checking and processing patients. For example, Alexandra hospital in Singapore adopted an RFID tracking system to monitor all patients, visitors, and staffs entering the hospital during the severe acute respiratory syndrome (SARS) outbreak in 2003. If someone was diagnosed with SARS later, all individuals who contacted the person in the hospital could be immediately identified.

An embedded chip is used as a core to establish a body area network in the prosthesis training system in this work. The 3D

positioning information of the amputee's limbs is acquired through installing RFID reader in the limb prosthesis. A gait pattern classifier is adopted to compare current gait pattern of the amputee with the sample gait pattern records in the database. Proper feedback messages are then issued to the leg-amputees in order to provide completed and effective gait training.

This article is organized as follows: the next section presents the proposed prosthesis training system. The simulation results are given in Section 3, and the final section concludes the work.

2. Architecture of intelligent prosthesis training system

The architecture of the proposed prosthesis training system is shown in Fig. 1. Several RFID readers are installed on the limb prosthesis as shown in Fig. 1(a). The system keeps receiving data from RFID tags attached on the railing at the both sides of the leg-amputee during gait training processing. Some proper feedback messages will be issued to the leg-amputee for correcting his/her gait.

As illustrated in Fig. 1(b), the feedback messages are generated by the four modules according to the received data from RFID tags in the system. First, RFID positioning module positions the gait of amputee based on real-time positioning information that RFID readers receive from tags on the railing. The training gait pattern recording module then records the angle variation of thigh and knee of leg-amputee with respect to time and the trainee's position. The gait pattern classification module is responsible for comparing current gait record of the amputee, which is forwarded by the training gait pattern recording module, with the sample gait patterns established in the system and assists in diagnosing whether the amputee would suffer from an accident during gait training. The feedback message module determines whether the system should issue a real-time guidance message to the amputee for correcting her/his gait according to the gait pattern classification result. Notably, the system will forward the amputee's gait record to the experts if the gait pattern mismatches all the samples in the database so as to obtain appropriate guide messages for the new gait patterns. The detailed descriptions of the above-mentioned four modules are given below.

2.1. RFID positioning module

Received signal strength (RSS) is adopted as the parameter of the RFID positioning algorithm to estimate the distance between the sender and the receiver. The detected signal strength decreases with the increasing distance to the sender. Thus, the detected signal will differ when the receiver is at different location. It can be shown that the remaining power of wave receiver in the signal propagation can be expressed as follows:

$$P_R = P_T \cdot G_T \cdot G_R \cdot \left(\frac{\lambda}{4\pi D_{is}} \right)^2, \tag{1}$$

where P_T is the transmission power of sender, G_T is the gain of transmitter, G_R is the gain of receiver, λ is the wave length, and D_{is} is the distance between sender and receiver. Notably, all the parameters are known and measurable except the distance.

The distance is thus determined by,

$$D_{is} = \left| \sqrt{\frac{P_T \cdot G_T \cdot G_R \cdot \lambda^2}{P_R \cdot (4\pi)^2}} \right|. \tag{2}$$

The railing and tag screen employed in the gait training of leg-amputee is shown in Fig. 2(a1). The RFID reader installed in the lower-limb prosthesis, as illustrated in Fig. 2(a2), can read several RFID tag data simultaneously. In the meantime, the locations of RFID reader and the two tags shown in Fig. 2(b) will form
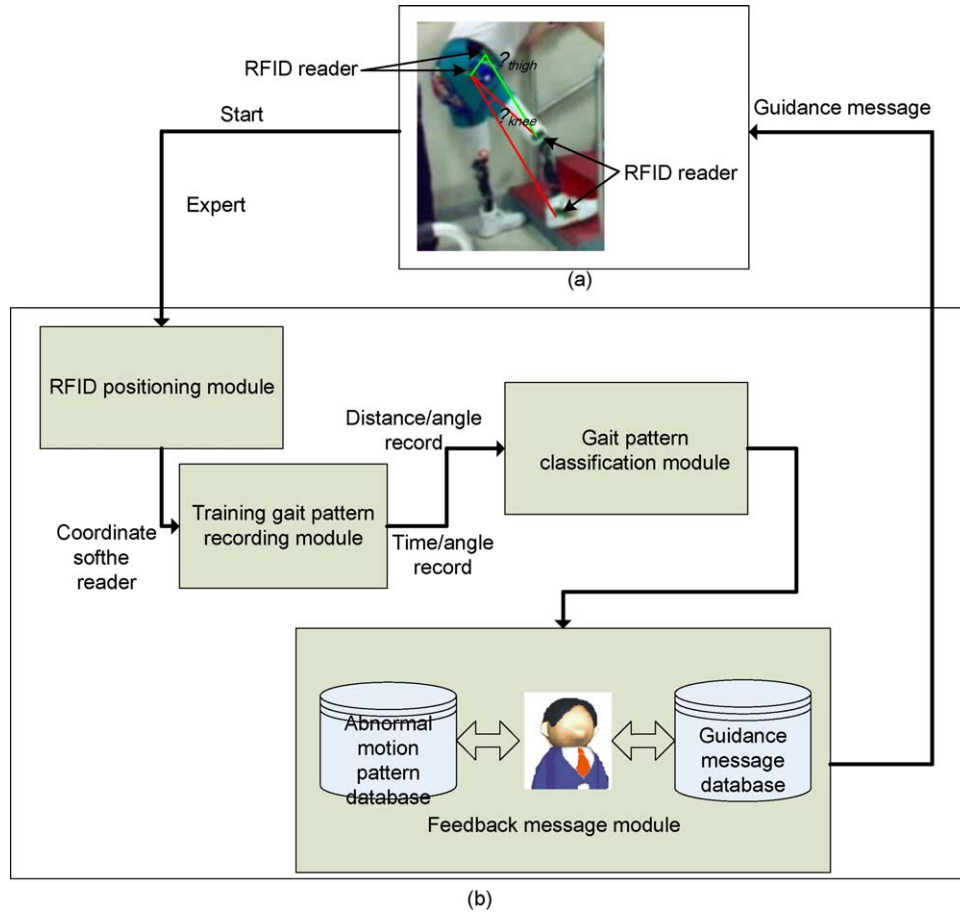
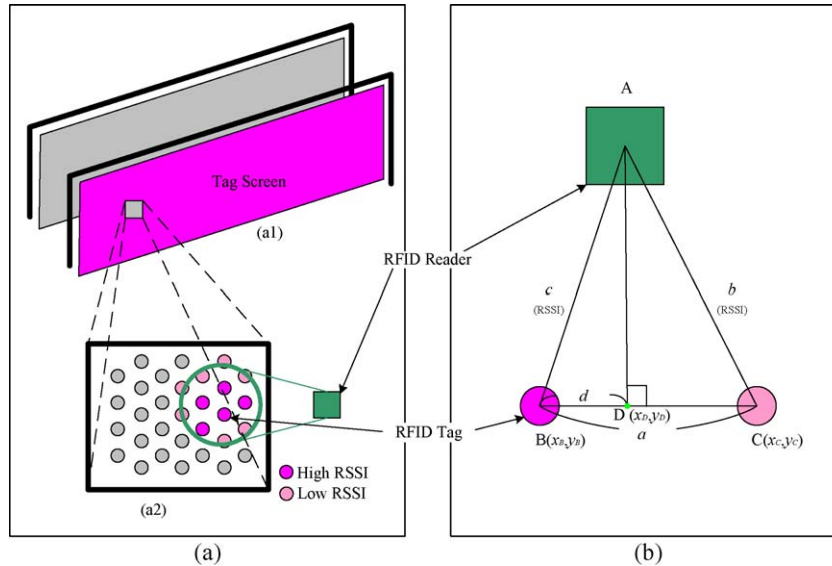Fig. 1. Architecture of the proposed prosthesis training system.

Fig. 2. Operation of RFID positioning module.

a triangle. The length a as given in Fig. 2(b) denotes the spacing between two RFID tags and it is fixed and set beforehand. The lengths b and c can be derived from signal strength of RSSI. $D(x_D, y_D)$ represents the coordinate of RFID reader projected on the bottom line of the triangle, and d can be derived accordingly by means of Pythagoras theorem. After all RFID readers are positioned, the training gait pattern recording module will take the

responsibility for recording the variation of gait patterns of the amputee.

$D(x_D, y_D)$ is as follows:

$$\begin{cases} x_D = (x_B + (x_C - x_B) \times \dfrac{d}{a} \\ y_D = (y_B + (y_C - y_B) \times \dfrac{d}{a} \end{cases} \qquad (3)$$
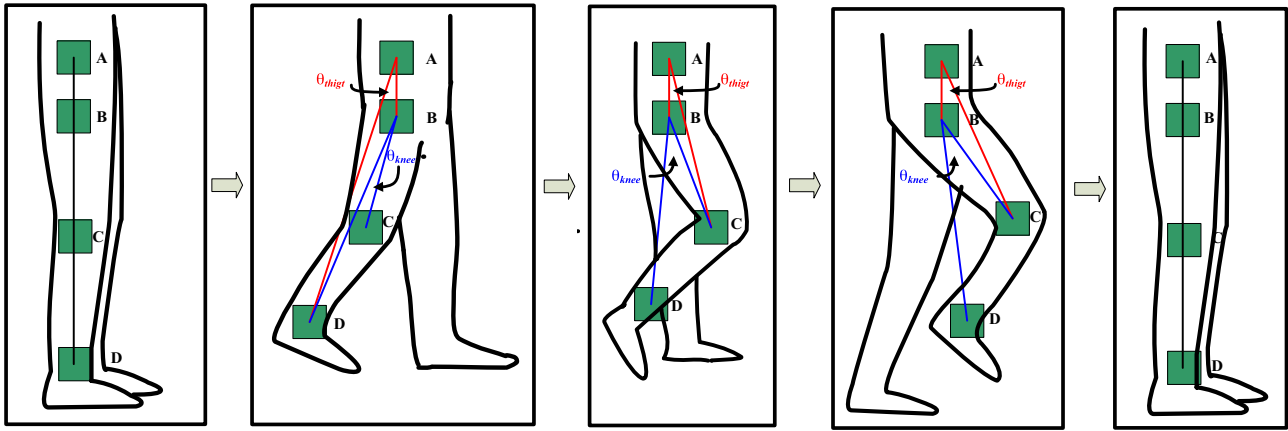
Fig. 3. Illustration of relative position variation of RFID readers while the trainee moves forward.

2.2. Training gait pattern recording module

Fig. 3 illustrates the relative position variation of the RFID readers while the trainee moves forward during the leg-amputee's gait training process. The angle variation of thigh and knee of leg-amputee with respect to time and the trainee's position will be recorded for further processing by the training gait pattern classifying module and the evaluation metric computation module. The angle variation of thigh and knee of leg-amputee can be derived by,

$$
\begin{aligned}
\theta_{\text{thigh}} &= \cos^{-1}\frac{\vec{BA} \cdot \vec{CA}}{|\vec{BA}||\vec{CA}|} \\
\theta_{\text{knee}} &= \cos^{-1}\frac{\vec{DB} \cdot \vec{CB}}{|\vec{DB}||\vec{CB}|}
\end{aligned}
\tag{4}
$$

where θ_{thigh} is angle variation of thigh, θ_{knee} denotes angle variation of knee, and A, B, C, and D represent four RFID readers installed in the prosthesis, respectively.

2.3. Gait pattern classification module

This module aims to classify the training tracks into different types of gait patterns. The red curves in Fig. 4 give four examples of the abnormal gait records for θ_{thigh} and θ_{knee}, from which we can derive the initial angle λ, the length of perpendicular H, and time consumption per step t as illustrated in the figure (For interpreta-

tion of the references to color in this sentence, the reader is referred to the web version of the article). Take Fig. 4(a) as an example, in which larger initial angle λ and smaller length of perpendicular H than those of the normal patterns is observed. This situation occurs while the trainee walks too fast. During prosthesis training, the module receives a new gait record and then records it in the table, Record T, where we can obtain three parameters to train the classification module. Decision tree and fast learning neural-networks (FLNNs) are adopted as the classification algorithms in this study. The motivation of using decision tree and neural networks here is that both techniques are powerful and popular tools for classification and prediction [14–19], and there are lots of solutions on VLSI chips which allow both techniques to be hardware-computed.

2.3.1. Decision tree

Decision tree [14–16] is a predictive mode, a mapping of observations about an item to conclude about the item's target value. Each interior node corresponds to a variable; an arc to a child represents a possible value of that variable. A leaf represents the predicted value of target variable given the values of the variables represented by the path from the root. There are several advantages for decision tree. For instance, it is simple to understand and interpret, and it is able to handle nominal and categorical data and perform well with large data set in a short time. In this work, we use C4.5 decision tree to predict the direction change of stock price
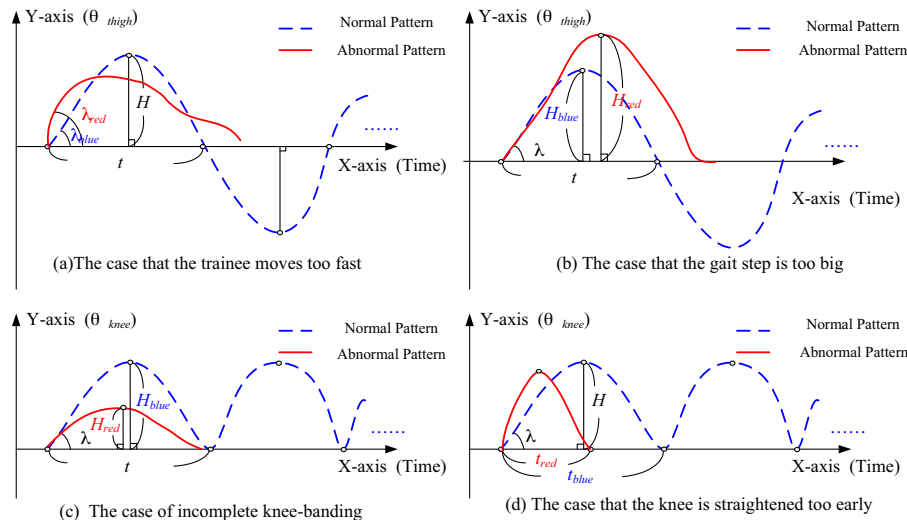

Fig. 4. Examples for abnormal gait pattern classification.

because C4.5 decision tree performs well in prediction application as report in [20].

A decision tree is a hierarchy of yes/no questions in which the specific questions asked depend on the answers given to the previous questions, with the branches spreading out from the original question until an appropriate response is given. Decision trees can be used to encapsulate the knowledge of an expert about a specific system. Various methods exist for the development of decision trees from datasets, with the goal of each method being to produce a structure that gives the highest degree of accuracy for the smallest tree design [21–23].

Decision tree learning is one of the most widely used and practical methods for inductive inference. It is a method for approximating discrete-valued functions that is robust to noisy data and capable of learning disjunctive expressions. C4.5, one of the most widely used decision tree learning algorithm, is adopted in our work.

In the process of constructing the decision tree, the root node is first selected by evaluating each attribute using a statistical test to determine how well it alone classifies the training examples. The best attribute is selected and used to test at the root node of the tree. A descendant of the root node is created for each possible value of this selected attribute, and the training examples are sorted to the appropriate descendant node. The entire process is then repeated using the training examples associated with each descendant node to select the best attribute to test at that point in the tree.

There are two frequently used metrics for attributes selection. One is the information gain, Gain(S, A) of an attribute A, relative to a collection of examples S,

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Value}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v), \qquad (5)$$

where Value(A) is the set of all possible values for attribute A, and S_v is the subset of S for which attribute A has value v. Entropy(S) is the entropy of S. The entropy is defined as:

$$\text{Entropy}(S) = -\sum_{t=1}^{c} p_i \log_2 p_i, \qquad (6)$$

where c denotes the number of all possible values for attribute A. Notably, the first term on the right hand side of Eq. (5) denotes the entropy of the original collection S, and the second term is the expected value of the entropy after S is partitioned using attribute A.

The second metric commonly used for attributes selection is called Gain ratio,

$$\text{Gain ratio}(S, A) = \frac{\text{Gain}(S, A)}{\text{Split information}(S, A)}, \qquad (7)$$

where Split information is defined as:

$$\text{Split information}(S, A) = \sum_{t=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}, \qquad (8)$$

where S_1 through S_c are the c subsets of examples resulting from partitioning S by the c-values attribute A. Note that Split information is actually the entropy of S with respect to the values of attribute A. Notably, the metric for attribute selection as given in Eq. (7) is employed in C4.5 algorithm for better performance achievement [24].

Fig. 5 demonstrates an example of decision tree constructed with three parameters, including the initial angle, the length of perpendicular, and time consumption per step. The example shown in Fig. 5 may happen while the trainee walks too fast or the movement of the prosthesis is too large.
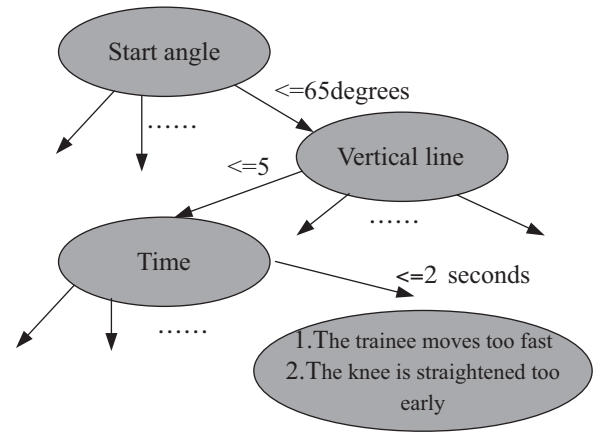

Fig. 5. An example of building decision tree in this work.

2.3.2. Fast learning neural-networks

The fast learning neural networks (FLNNs), as shown in Fig. 6, are chosen owing to their superior performance in the literature [25]. The major problem of applying neural networks is that it is not impractical to use neural networks in real-time applications because it takes too much time during neural network training process. However, there have been lots of solutions on VLSI chips which allow the neural networks to be hardware-computed, thus the implementation of neural networks by hardware becomes feasible nowadays. Besides, this work replaces the classical back propagation learning by a fast learning algorithm to reduce the overlong training time that usually occurs in the traditional neural network models.

The FLNNs [25] which adopt the fast learning algorithm can automatically select appropriate value of neural quantizers and analytically determine the weights and bias of the network at one time. This kind of neural-networks has the following characteristics:

(1) The neural networks are composed of two hidden layers, with first larger and second narrower. There are several neuron quantizers that connect to the input layer and one or several neurons in the second hidden layer.
(2) All the connection weights between the input layer and the first hidden layer can be simply prefixed, and most of them are randomly assigned.
(3) The connection weights between the first hidden layer and second hidden layer can be determined at one time, instead of the iterative adjustment adopted in traditional neural network learning algorithms.
(4) The connection weights between the second hidden layer and the output layer can be simply set as a constant value C.
(5) Weight size factor wf or C and quantizer factors qt and qu should be adjusted to optimize generalization performance.

The fast learning algorithm (FLA) adopted in this work can be summarized as follows:
Inputs of FLA:

1. N arbitrary distinct samples (x_i, t_i), where $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in \Re^n$ and $i = 1, ..., N$.
2. Expected learning accuracy $\varepsilon < 0$.
3. Number of groups G.
4. Number of neurons D of the first hidden layer.

The learning procedure of FLA:
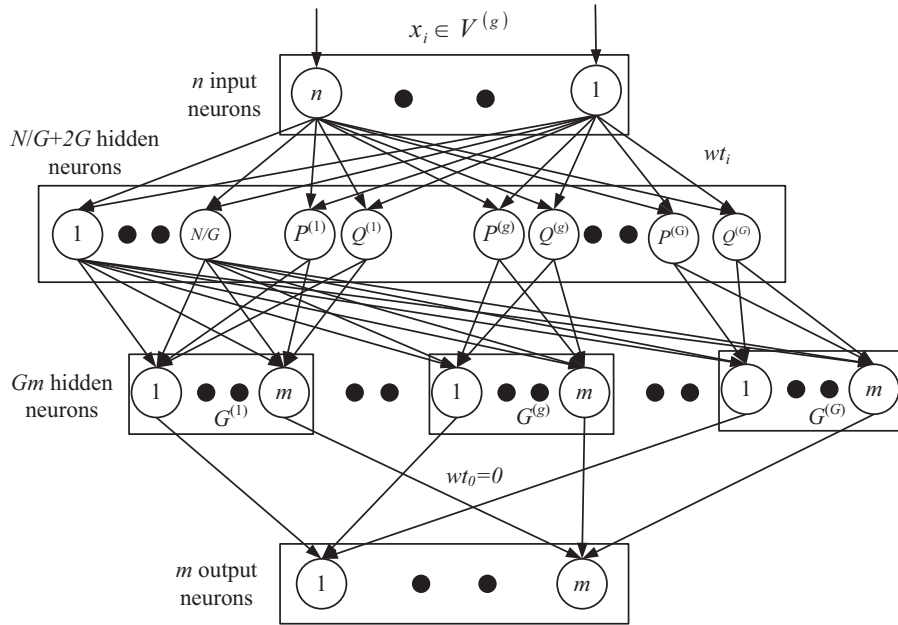*Step* 1: Sorting and grouping inputs.

Fig. 6. Neural-network with fast learning algorithm.

(a) Randomly choose vector $wt \in \Re^n$ and re-index inputs such that $wt \cdot x_1 < wt \cdot x_2 < \cdots < wt \cdot x_N$.

(b) Group sorted inputs into G groups $V^{(g)}$, $g = 1 \sim G$

$$V^{(g)} = \left\{ x_i \,\middle|\, wt \cdot x_{(g-1)N/G+1} \leq wt \cdot x_i \leq wt \cdot x_{gN/G} \right\}. \qquad (9)$$

Step 2: Determination of weights and biases of two hidden layers.

(a) Randomly choose the weights wt_i and biases b_i, where $i = 1, ..., D$, and $D \leq N/G$ is the number of neurons in the first hidden layer.

(b) Choose $C = \alpha \cdot \max\limits_{1 \leq i \leq n; 1 \leq j \leq m} |t_{ij}|$, where α can be any positive number larger than two.

(c) Calculate matrix $\gamma^{(g)} = [\gamma_1^{(g)}, \gamma_2^{(g)}, ..., \gamma_D^{(g)}]^T$:

$$\gamma^{(g)} = (M^{(g)})^* U^{(g)} \gamma^{(g)} = (M^{(g)}), \qquad (10)$$

$$M^{(g)} = \begin{bmatrix} \sigma(wt_1 \cdot x_{(g-1)N/G+1} + b_1) & \cdots & \sigma(wt_D \cdot x_{(g-1)N/G+1} + b_D) \\ \vdots & \cdots & \vdots \\ \sigma(wt_1 \cdot x_{gN/G} + b_1) & \cdots & \sigma(wt_D \cdot x_{gN/G} + b_D) \end{bmatrix}, \qquad (11)$$

where $\sigma(x) = 1/(1 + e^{-x})$, and $(M^{(g)})^*$ stands for the Moore–Penrose generalized inverse of $(M^{(g)})$.

$$U^{(g)} = \begin{bmatrix} \ln\left(\dfrac{0.5 + t_{(g-1)N/G+1,1}/C}{0.5 - t_{(g-1)N/G+1,1}/C}\right) & \cdots & \ln\left(\dfrac{0.5 + t_{(g-1)N/G+1,m}/C}{0.5 - t_{(g-1)N/G+1,m}/C}\right) \\ \vdots & \cdots & \vdots \\ \ln\left(\dfrac{0.5 + t_{gN/G,1}/C}{0.5 - t_{gN/G,1}/C}\right) & \cdots & \ln\left(\dfrac{0.5 + t_{gN/G,m}/C}{0.5 - t_{gN/G,m}/C}\right) \end{bmatrix}. \qquad (12)$$

Step 3: Determination of weights and biases of neural quantizers.

(a) Set the quantizers factor qt and qu as

$$qt = \frac{2\ln\left(2qu/\min\limits_{i=1 \sim N} \ln\left(\frac{(C+\varepsilon/\sqrt{m}-2t_{ij})(C+2t_{ij})}{(C-\varepsilon/\sqrt{m}+2t_{ij})(C-2t_{ij})}\right) - 1\right)}{\min\limits_{j=1 \sim N-1}(wt \cdot x_{j+1} - wt \cdot x_j)}, \qquad (13)$$

$$qu = \ln\left(\frac{2\sqrt{m}CG}{\varepsilon} - 1\right) + \max\limits_{\substack{g=1 \sim G \\ q=1 \sim N/G \\ s=1 \sim G}} \left\| M_q^{(g)} \cdot \gamma^s \right\|_\infty$$

$$+ \min\limits_{i=1 \sim N} \ln\left(\frac{(C + \varepsilon/\sqrt{m} - 2t_{ij})(C + 2t_{ij})}{(C - \varepsilon/\sqrt{m} + 2t_{ij})(C - 2t_{ij})}\right). \qquad (14)$$

(b) Set the connection weights $wt_{P(g)}$ and wt_{Q^g} that link the input layer and neurons $P^{(g)}$ and $Q^{(g)}$, $g = 1, 2, ..., G$, as

$$\begin{aligned} wt_{p(g)} &= qt \cdot wt \\ wt_{Q(g)} &= -qt \cdot wt \end{aligned}. \qquad (15)$$

(c) Set the biases of neurons $P^{(g)}$ and $Q^{(g)}$, $g = 1, 2, ..., G$ as

$$\overline{b}_{p(g)} = \begin{cases} -qt\left(\dfrac{1}{2}wt \cdot x_{gN/G} + \dfrac{1}{2}wt \cdot x_{gN/G+1}\right), & \text{if } g \neq G \\ -qt(wt \cdot x_N + \max\limits_{j=1 \sim N-1}(wt \cdot x_{j+1} - wt \cdot x_j)), & \text{if } g = G \end{cases}, \qquad (16)$$

$$\overline{b} = \begin{cases} qt\left(\dfrac{1}{2}wt \cdot x_{(g-1)N/G+1} + \dfrac{1}{2}wt \cdot x_{(g-1)N/G}\right), & \text{if } g \neq 1 \\ qt(wt \cdot x_1 + \max\limits_{j=1 \sim N-1}(wt \cdot x_{j+1} - wt \cdot x_j)), & \text{if } g = 1 \end{cases}. \qquad (17)$$

(d) Set the connection weights w_{AB} that link neurons $A^{(p)}$ and $B^{(p)}$ and the second hidden layer as

$$w_{PQ} = -qu. \qquad (18)$$

(e) Set the connection weights and the biases that link neurons of the second hidden layer and the output layer as

$$wt_o = C. \qquad (19)$$

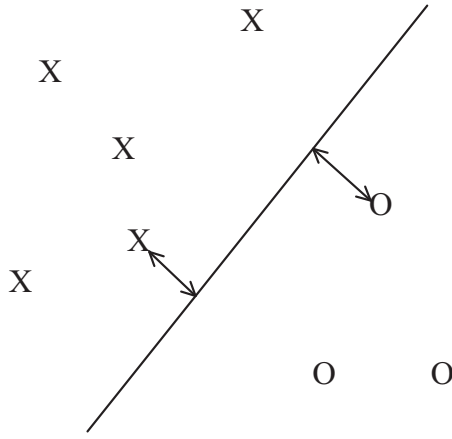$$\overline{b}_o = -0.5C, \qquad (20)$$

Fig. 7. Support vectors and margin.

Fig. 8. A screen shot of feedback message provided by the system.

2.3.3. Support vector machines

In a decade, SVMs [26–30] have attracted much attention as a new classification technique with good generalization ability. The basic idea of SVMs is to map input vectors into a high-dimensional feature space and linearly separate the feature vectors with an optimal hyper-plane in terms of margins, i.e. distances of given examples from a separating hyper-plane.

SVMs are promising methods for the prediction of financial time series because they use a risk function consisting of the empirical error and a regularized term which is derived from the structural risk minimization principle. Given a training dataset represented by the X-matrix $(X_1, ..., X_m)$ divided into two linearly separable classes with class labels (+1 and −1) stored in the Y-vector $(Y_1, ..., Y_m)$ as given in Fig. 7, the maximum margin plane can be found by minimizing ($\|w\|_2$):

$$\|w\|_2 = w \cdot w = \sum_{i=1}^{d} w2i \tag{21}$$

with constraints:

$$yi(w \cdot x_i + b) \geq 1, \tag{22}$$

where $i = 1, ..., m$, $b \in R$, and $x_i \in R^d$.

Fig. 7 shows the simplest case of two linearly separable classes. The concept of support vectors is illustrated by the points closest to the surface separating two classes, and the margin is given by the distance between support vectors and separating surface.

The decision function takes the form $f(x) = sgn(w \cdot x + b)$, where sgn($\cdot$) is simply a sign function which returns +1 for positive arguments and −1 for negative arguments. This simple classification problem is generalized to a non-separable case by introducing slack variables x_i and minimizing the following quantity:

$$\frac{1}{2} w \cdot w = C \sum_{i=1}^{m} \xi_i, \tag{23}$$

where $y_i(w \cdot x_i + b) \geq 1 - \xi_i$ and $\xi_i > 0$.

The above quadratic optimization problem with constraints can be reformulated by introducing Lagrangian multipliers α, v, and the following Lagrangian is formed,

$$L(w, b, \xi, \alpha, v) = \frac{1}{2} w \cdot w + C \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i[y_i(w \cdot x_i + b) - 1 + \xi_i]$$
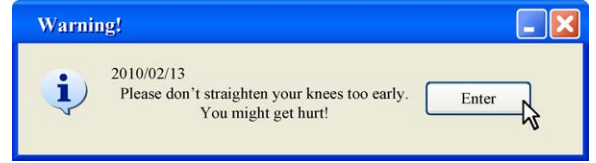$$- \sum_{i=1}^{m} v_i \xi_i \tag{24}$$

Stationary points of this Lagrangian can be obtained by:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0, \tag{25}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0, \tag{26}$$

$$\frac{\partial L}{\partial \xi_i} = \alpha_i + v_i - C = 0. \tag{27}$$

Two remaining derivatives $((\partial L/\partial \alpha), (\partial L/\partial v))$ recover the constraint equations. By substituting the expression $w = \sum_{i=1}^{m} \alpha_i y_i x_i$ back into the Lagrangian, we obtain this simpler dual formulation

$$W(\alpha) = \frac{1}{2} \sum_{i=1}^{m} \alpha_i \alpha_j y_i y_j x_i \cdot x_j - \sum_{i=1}^{m} \alpha_i, \tag{28}$$

with the constraints $0 \leq \alpha_i \leq C$ and $\sum \alpha_i y_i = 0$.

2.4. Given a mapping

$$x \rightarrow \phi(x), \tag{29}$$

the dot product in the final space can be replaced by a Mercer kernel

$$\phi(x)\phi(y) \rightarrow K(x, y). \tag{30}$$

Since $\phi(\cdot)$ typically maps x into the space of much higher dimensionality, it is usually specified by defining the kernel implicitly. The above dual formulation thus becomes a preferred approach due to highly dimensional feature space induced by $\phi(\cdot)$ mapping. The decision function for classification problems is then given by:

$$f(x) = sgn\left(\sum_{sv} \alpha_i y_i K(x_i, x) + b\right). \tag{31}$$

2.5. Feedback message module

This module provides the trainees appropriate feedback messages by means of a rule-based expert system that includes the trajectory pattern database and the feedback message database. Sample gait patterns are collected in the trajectory pattern database during gait training. The gait record of the amputee is compared with the sample gait patterns in the trajectory pattern database, and a feedback message can then be generated from the feedback message database depending on how long the gait training given to the trainee. For example, when the system detects that the trainee's gait step is too large, the system will deliver a message, "Please slow down your movement", for the beginner because a trainee tends to be nervous at first. On the other hand, a message such as, "Please don't straighten your knees too early. You might get hurt!", will be offered if a well-trained user walks too fast. Fig. 8 shows a screen shot of an example of the feedback message provided by the feedback message module. Notably, if a new gait sample cannot be judged by the feedback message module, it will be added in the feedback message database

Table 1
Classification results.

Algorithm	Rate	RMSE	Dev
Decision tree	88.9254%	0.117	1.83%
FLNNs	89.0351%	0.0642	0.64%
SVM	89.2544%	0.2109	9.11%

Table 2
Computation time comparison of three classifiers.

Algorithm	Training time (s)	Testing time (s)
Decision tree	0.05	0.02
FLNNs	0.007	0.003
SVM	5.91	4.33

with the assistance of the expert. Moreover, the expert can add appropriate feedback messages according to the trainees' portfolio database.

3. Simulation results

In the experiments conducted in this research, 27 kinds of gait patterns were identified by experienced experts who specialize in prosthesis training. 912 gait training patterns were collected from the leg-amputees during real gait training process and were classified by an expert. The classification accuracy ratio was computed by using the leave one out cross validation (LOOCV) technique [31] to examine the effectiveness of the proposed training system. Notably, the LOOCV method removes a single sample in each trial, trains on the rest, and then tests the classifier on the removed single sample.

The classification results are given in Table 1. It can be seen that, in average, accuracy ratio of the proposed training system is 89.0716%, the testing root mean square error (RMSE) is 0.1307, and the its standard deviation (DEV) is 3.86%. It implies that the proposed prosthesis training system not only detects the movements of leg-amputees, but also classifies most of the gait patterns correctly. Meanwhile, the performance of the three compared classifiers is about the same. Notably, the computation time comparison for the three classifiers, as shown in Table 2, reveals that the computation time required for both the training and the testing stage is insignificant for FLNNs. The simulation results thus exhibit the effectiveness and practicability of the proposed prosthesis training system.

4. Conclusions

A prosthesis training system that utilizes an angular comparison method is proposed in this work to replace the traditional graphic comparison method owing to the requests for automatic and real-time guidance. A so-called dissimilarity degree is defined in this work to diagnose abnormal gait patterns that the trainee might suffer from. Instant and adaptive feedback messages that are useful to correct their walking postures are given by the prosthesis training system during the self-training process. An evaluation metric is presented in this work to assist in providing suitable messages to the trainees under different training stages. A series of simulations were conducted in order to evaluate the performance of the proposed system. The simulation results reveal the effectiveness and practicability of proposed prosthesis training system.

The economic factor usually determines whether a leg-amputee can accept complete and sufficient gait training. The cost of the proposed system is affordable to the trainees owing to wide deployment of the low-cost RFIDs. In addition, the new gait samples can be added in the database with the assistance of the experts if a mismatch occurs between the training pattern and the samples recorded in the database. Therefore, the prosthesis training system proposed this work can greatly benefit the amputees because they can borrow or buy from the limb prosthesis vendors and train themselves at home. The future work will focus on developing more advanced algorithms in order to improve the classification accuracy of the proposed work.

Acknowledgments

References

[1] International classification of diseases, ninth revision (ICD-9) principal procedure code 84.1, 2005 HCUPnet. Healthcare Cost Utilization Project (HCUP). Available from: http://hcupnet.ahrq.gov/ (online).
[2] D.A. Winter, S.E. Sienko, Biomechanics of below-knee amputee gait, J. Biomech. 21 (5) (1988) 361–367.
[3] H.B. Skinner, D.J. Effeney, Gait analysis in amputees, Amer. J. Phys. Med. 64 (1985) 82–89.
[4] H. Bateni, S. Olney, Kinematic and kinetic variations of below-knee amputee gait, J. Prosthet. Orthot. 14 (1) (2002) 2–13.
[5] M.S. Orendurff, A.D. Segal, G.K. Klute, M.L. McDowell, J.A. Pecoraro, J.M. Czerniecki, Gait efficiency using the c-leg, J. Rehabil. Res. Develop. 43 (2) (2006) 239–246.
[6] A.F.T. Mak, M. Zhang, D.A. Boone, State-of-the-art research in lower-limb prosthetic biomechanics – socket interface, J. Rehabil. Res. Develop. 38 (2001) 161–167.
[7] H.M. Azamathulla, A.A. Ghani, C.K. Chang, Z.A. Hassan, N.A. Zakaria, Machine learning approach to predict sediment load – a case study, Clean – Soil, Air, Water (2010).
[8] H.M. Azamathulla, W. Fu-Chun, A.A. Ghani, S. Narulkar, N.A. Zakaria, C.C. Kiat, Comparison between genetic algorithm and linear programming approach for real time operation, J. Hydro-Environ. Res. (2008) (Elsevier & KWRA).
[9] S.K. Au, J. Weber, H. Herr, Powered ankle–foot prosthesis improves walking metabolic economy, IEEE Trans. Robot. 25 (1) (Feb. 2009) 51–66.
[10] "Bluetooth: The Next Advancement for Prosthetics," internet journal of emerging medical technologies, 2008. Available from: http://medgadget.com/archives/2008/01/bluetooth_the_next_advancement_for_prosthetics.html.
[11] R. Want, A. Hopper, V. Falcão, J. Gibbons, The active badge location system, ACM Trans. Inform. Syst. 10 (1) (Jan. 1992) 91–102.
[12] J. Arias, A. Zuloaga, J. Lázaro, J. Andreu, A. Astarloa, Malguki: an RSSI based ad hoc location algorithm, Microprocess. Microsyst. 28 (8) (Oct. 4 2004).
[13] L.M. Ni, Y. Liu, Y.C. Lau, A.P. Patil, LANDMARC: indoor location sensing using active RFID, in: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 23–26 March 2003, 2003, pp. 407–415.
[14] G. Hughes, On the mean accuracy of statistical pattern recognizers, IEEE Trans. Inform. Theory 14 (1) (1968) 55–63.
[15] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Chapman and Hall/CRC, 1984.
[16] S.P. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE Trans. Syst. Man Cybern. 21 (3) (1991) 660–674.
[17] H.M. Azamathulla, A.A. Ghani, N.A. Zakaria, C.K. Chang, C.S. Leow, Knowledge extraction from trained neural network scour models, Mod. Appl. Sci. (2008).
[18] H.M. Azamathulla, A.A. Ghani, N. Azazi Zakaria, ANFIS-based approach to predicting scour location of spillway, Water Manage. J. 162 (6) (2009) 339–407.
[19] H.M. Azamathulla, A.A. Ghani, An ANFIS-based approach for predicting the scour depth at culvert outlet, J. Pipeline Syst. Eng. Pract. ASCE (2011).
[20] X. Zhu, J. Wang, H. Yan, S. Wu, Research and application of the improved algorithm C4.5 on decision tree, Int. Conf. Test Measur. 2 (2009) 184–187.
[21] T. Endou, Q.F. Zhao, Generation of comprehensible decision trees through evolution of training data, IEEE Congr. Evol. Comput. 2 (2002) 1221–1225.
[22] X. Llora', J.M. Garrell, Evolution of decision trees, in: Conference on Artificial Intelligence, 2001.
[23] A. Papagelis, D. Kalles, Breeding decision trees using evolutionary techniques, in: International Conference On Machine Learning, Williams College, Williamstown, MA, USA, 2001.
[24] R.J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.
[25] G.B. Huang, Q.Y. Zhu, C.K. Siew, Real-time learning capability of neural networks, IEEE Trans. Neural Netw. 17 (4) (2006) 863–878.
[26] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.
[27] N. Cristianini, J.S. Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, 2000.

[28] C.J. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov. 2 (1998) 121–167.

[29] S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan College Publishing Company, New York, 1994.

[30] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, 2001. Software available from: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[31] G. Earl, R. Johnsonbaugh, S. Jost, Pattern Recognition and Image Analysis, Prentice Hall, 1996.