

美和科技大學

100 年度教師產學合作計畫 結案報告

計畫名稱：跌倒偵測系統之開發

計畫編號： 100-FI-DHIIT-IAC-R-004

計畫期間：自民國 100 年 4 月 1 日起至民國 100 年 12 月 31 日止。

計畫主持人： 游義地

共同主持人：

研究助理：

經費總額： 50,000 元

經費來源： 首羿國際股份有限公司

摘 要

跌倒所引起的傷害，對於老年人居家照護而言是一項重要的防護工作。由於老年人的體質較年青人虛弱，跌倒所引發的傷害，甚至會影響日常生活行為。本研究的目的，是開發可攜帶的智慧型跌倒偵測系統，降低老年人跌倒所造成的傷害。本系統利用三軸加速器量測老年人活動之 X、Y、Z 三軸加速度值，再由結合類神經網路之晶片判讀是否為跌倒狀態；若為跌倒，則將訊號傳至後端跌倒偵測系統通知照護人員立即處理。為了提高本系統的利用價值，所開發的攜帶設備具有 GPS 定位功能，老年人於戶外活動時可將經、緯度座標值傳回管理平台，與 Google Map 結合可立即定位出老年人所在位置，提高老年人的行動安全。

壹、前言

我國自82年起邁入高齡化社會以來，65歲以上老人所占比例持續攀升，99年底已達10.7%，老化指數為68.6%，近3年間就大升10.5個百分點。老年人目前獨居或僅與配偶同住者比例合占27.9%，住在安養或養護機構者僅占2.8%。經我國內政部的調查，老人對老年生活的期望前三項皆依序為「身體健康的生活」、「能與家人團圓和樂的生活」及「經濟來源無虞的生活」。對老年生活擔心的問題前三項皆依序為「自己的健康問題」、「經濟來源問題」、「自己生病的照顧問題」[1]。

由以上的資料顯示，安全的健康生活是老年人所渴望的。近年來有許多的案例指出，有不少的獨居老年人因跌倒未被發現，而延後治療造成嚴重的傷害，甚至危及生命。隨著科技的發展，有許多的學者利用三軸加速器、陀螺儀、影像監視系統或GPS等儀器設備開發跌倒偵測系統[2~4]。跌倒偵測系統可分為固定式、攜帶式及混合式三種，固定式一般是以影像監視為主，利用影像判讀老年人活動姿態，由於活動範圍受限制且關係到個人隱私權，因此在使用上須格外謹慎；攜帶式通常是配帶在老年人身上，當偵測到跌倒狀態時，透過無線網路、Zigbee或藍芽等無線通訊將訊號傳至管理中心；混合式則結合監視系統及隨身裝置，成本較高。

本研究的目的是，是開發可攜帶的智慧型跌倒偵測系統，降低老年人跌倒所造成的傷害。本系統利用三軸加速器量測老年人活動之 X、Y、Z 三軸加速度值，再由結合類神經網路之晶片判讀是否為跌倒狀態；若為跌倒，則將訊號傳至後端跌倒偵測系統通知照護人員立即處理。為了提高本系統的利用價值，所開發的攜帶設備具有 GPS 定位功能，老年人於戶外活動時可將經、緯度座標值傳回管理平台，與 Google Map 結合可立即定位出老年人所在位置，提高老年人的行動安全。

貳、研究方法

軟體開發環境

由於c#已廣泛運用在資訊平台應用軟體上，且程式設計相關資料十分完備，本系統程式所採用的語言為C# 2008版[5、6]，結合ACCESS資料庫，達到資訊儲存之目的，本系統軟體開發環境如表1所示。

表 1：軟體開發環境

開發環境	優點
開發語言 C#2008	選擇 C#作為開發工具的好處，是可以以極短的時間與成本製作出 Windows 各式應用軟體、資料庫軟體等。
執行環境 Windows XP	使用較普及

倒傳遞類神經網路

倒傳遞類神經網路(Back-propagation Neural Network)是一種具有學習能力的多層前授型網路。此種神經網路，是由 Rumelhart、McClelland 在 1985 年所提出的，當初他們之所以會提出此種網路，是希望提出一種平行分布訊息的處理方法來探索人類認知的微結構。

網路架構

倒傳遞網路的網路架構如圖 1 所示，包含了輸入層、隱藏層、輸出層；而以隱藏層可以不只一層。基本上，其網路架構於前授型多層感知機的網路架構類似，且每一層皆由一些神經元建構而成。請注意：在圖 1 中，同一層中的神經元彼此並不相連，而不同層間的神經元則彼此相連，且信號的流向是由輸入層向輸出層單向傳播。

倒傳遞網路中的神經元，其最常用的非線性轉換函數為雙彎取函數 (sigmoid function)，如圖 2 所示，其中

$$f(x) = \frac{1}{1 + e^{-x}}$$

這種函數有一種特性，即當 x 趨近於正負無窮大時， $f(x)$ 趨近於 0 或 1，而 $f(x)$ 的值則介於 (0, 1) 之間。

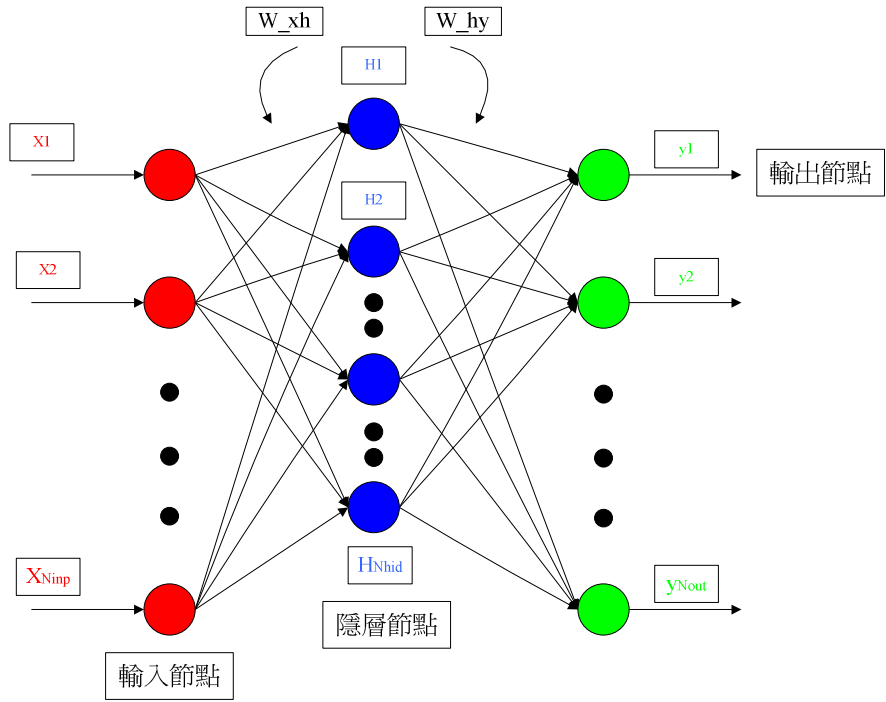


圖1：倒傳遞神經網路

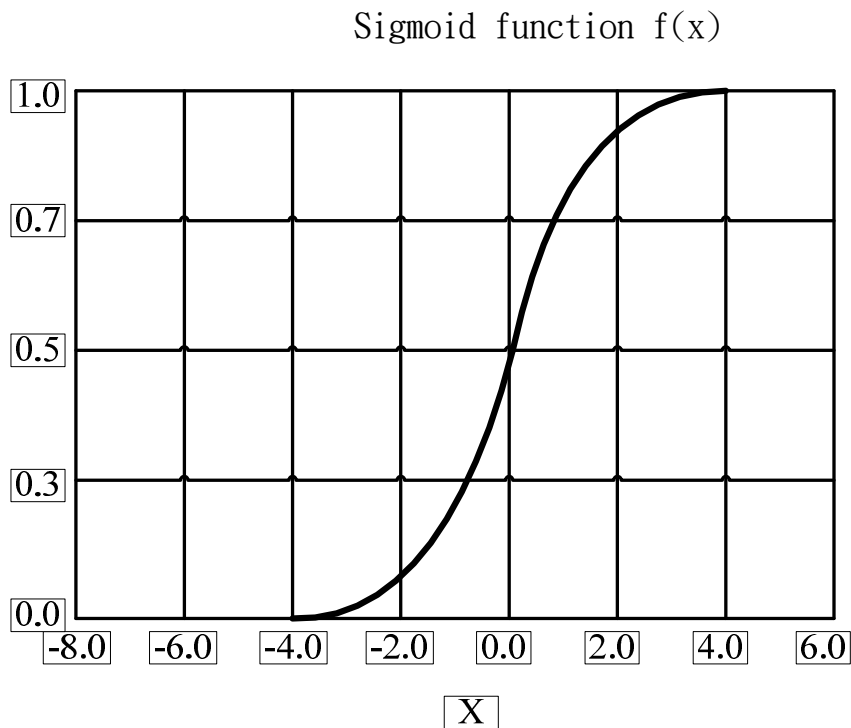


圖2：雙彎曲函數

倒傳遞網路演算法

倒傳遞網路演算法，簡稱為 BP 演算法，分為學習及回想兩部分，將這 BP 演算法的學習步驟及回想步驟詳細說明如下。

學習演算法

BP 學習演算法共分為 8 個步驟，說明如下：

步驟一：決定網路的層數及各層間神經元數目。

說明：為了讓讀者容易明瞭 BP 演算法，在此假設網路的架構為輸入

層、一層隱藏層及輸出層，即假設網路是三層網路架構，且假設輸入層的神經元數目有 N_{inp} 個、隱藏層的神經元數目有 N_{hid} 個，而輸出層的神經元數目有 N_{out} 個。

步驟二：以均佈隨機亂數設定網路的初始加權值及初始偏權值。

說明：因為不同層的神經元彼此相連，如果我們令 $W_{xh}[i][h]$ 為輸入層第 i 個神經元與隱藏層第 h 個神經元的加權值，由於有 N_{inp} 個輸入神經元與 N_{hid} 的隱藏神經元，所以我們可以用一個雙層迴圈來設定所有輸入層與隱藏層間的初始加權值，方法如下：

```
for i=1 to  $N_{inp}$ 
  for h=1 to  $N_{hid}$ 
     $W_{xh}[i][h]$  = 均佈隨機亂數
```

同理，如果我們令 $W_{hy}[h][y]$ 為隱藏層第 h 個神經元與輸出層第 j 個神經元間的加權值，則我們設定所有隱藏層與輸出層間的初始加權值的方法如下：

```
for h=1 to  $N_{hid}$ 
  for j=1 to  $N_{out}$ 
```

$W_{hy}[h][j]$ 均佈隨機亂數

在來我們要設定網路中的初始偏權值，要注意的是只有隱藏層及輸出層才有偏權值，輸入層是沒有的。事實上，輸入層是沒有運算能力的，他只是將一個神經元接收到的訊號平行輸出至隱藏層各個神經元中。若令 $\theta_h[h]$ 為隱藏層第 h 個神經元的偏權值， $\theta_y[j]$ 為輸出層第 j 個神經元的偏權值，則設定初始偏權值的方法如下：

for $h=1$ to N_{hid}

$\theta_h[h]$ = 均佈隨機亂數

for $j=1$ to N_{out}

$\theta_y[j]$ = 均佈隨機亂數

步驟三：輸入訓練樣本 $x[1], x[2], \dots, x[N_{inp}]$ 及目標輸出值

$T[1], T[2], \dots, T[N_{out}]$

說明：輸入值 $x[1], x[2], \dots, x[N_{inp}]$ 可為任意的實數值，但是由於倒傳遞網路採用雙彎曲函數(sigmoid function)當做神經元的非線性轉換函數，網路的推論輸出值的值域也必須落在 $[0, 1]$ 之間，所以目標輸出值 $T[1], T[2], \dots, T[N_{out}]$ 其值域也必須落在 $[0, 1]$ 之間。

步驟四：計算網路的推論輸出值 $Y[1], Y[2], \dots, Y[N_{out}]$ 。

說明：計算的方法是先算出隱藏層的輸出值，方法如下：

for h=1 to Nhid

$$net_h[h] = \sum_{i=1}^{Ninp} W_xh[i][h] \cdot x[i] - \theta_h[h] \quad (2-6)$$

for h=1 to Nhid

$$H[h] = \frac{1}{1 + e^{-net_h[h]}} \quad (2-7)$$

其中 $net_h[h]$ 為隱藏層第 h 個神經元的加權乘積和，而 $H[h]$ 為隱藏層第 h 個神經元的輸出值，他將收集到的加權乘積和 $net_h[h]$ 再做一次非線性轉換。

由於輸出層的輸入訊號來自隱藏層的輸出值，所以其推論輸出值可計算如下：

for j=1 to Nout

$$net_y[j] = \sum_{h=1}^{Nhid} W_hy[h][j] \cdot H[h] - \theta_y[j] \quad (2-8)$$

for j=1 to Nout

$$Y[j] = \frac{1}{1 + e^{-net_y[j]}} \quad (2-9)$$

其中 $net_y[j]$ 及 $Y[j]$ 分別是輸出層第 j 個神經元的加權乘積和及推

論輸出值。

步驟五：計算輸出層與隱藏層的差距量。

說明：計算輸出層差距量的公式如下：

for j=1 to Nout

$$\delta_y[j] = Y[j] \cdot (1 - Y[j]) \cdot (T[j] - Y[j]) \quad (2-10)$$

其中 $\delta_y[j]$ 是輸出層第 j 個神經元的差距量。在公式 $(T[j] - Y[j])$ 表示目標輸出值與網路推論輸出值間的誤差，所以 $\delta_y[j]$ 表示 $T[j]$ 與 $Y[j]$ 之間的誤差量度。

而計算隱藏層差距量的公式如下：

for h=1 to Nhid

$$\delta_h[h] = H[h] \cdot (1 - H[h]) \cdot \sum_{j=1}^{Nout} W_hy[h][j] \cdot \delta_y[j] \quad (2-11)$$

其中 $\delta_h[h]$ 表示隱藏層第 h 個神經元的差距量。請注意，在(2-11)

式中包含了一項子式：

$$\sum_{j=1}^{N_{out}} W_{hy}[h][j] \cdot \delta_y[j]$$

此式表示輸出層差距量的加權乘積和。所以 $\delta_h[h]$ 的計算與輸出層的差距量有關，這意味我們將輸出層的誤差倒傳至隱藏層來計算其差距量，這是此網路之所以有“倒傳遞”名稱的由來。

步驟六：計算各層間的加權值修正量及偏權值修正量。

說明：若令 $\Delta W_{hy}[h][j]$ 表示隱藏層第 h 個神經元與輸出層第 j 個神經元間的加權值修正量，且令 $\Delta \theta_y[j]$ 表示輸出第 j 個神經元的偏權值修正量，則計算其間所有加權值及偏權值修正量的方法如下：

for $h=1$ to N_{hid}

for $j=1$ to N_{out}

$$\Delta W_{hy}[h][j] = \eta \cdot \delta_y[j] \cdot H[h]$$

(2-12)

$$\Delta \theta_y[j] = -\eta \cdot \delta_y[j]$$

(2-13)

其中 η 為學習速率，一般取值為 0.1~1.0。寫時為了加速網路的收斂

速度，可將公式(2-12)與(2-13)改寫成

$$\Delta W_{hy}[h][j] = \eta \cdot \delta_y[j] \cdot H[h] + \alpha \cdot \Delta W_{hy}[h][j]$$

(2-14)

$$\Delta \theta_y[j] = -\eta \cdot \delta_y[j] + \alpha \cdot \Delta \theta_y[j]$$

(2-15)

其中 α 為慣性因子，一般取值為 0.0~0.9。

同理，若令 $\Delta W_{xh}[i][h]$ 為輸入層第 i 個神經元與隱藏層第 h 個神經元間上加權值修正量，且令 $\Delta \theta_h[h]$ 為隱藏層第 h 個神經元的偏權值修正量，則計算其間所有加權值及偏權值修正量的公式如下：

for $i=1$ to N_{inp}

for $h=1$ to N_{hid}

$$\Delta W_{xh}[i][h] = \eta \cdot \delta_h[h] \cdot x[i] + \alpha \cdot \Delta W_{xh}[i][h]$$

(2-16)

for $h=1$ to N_{hid}

$$\Delta \theta_h[h] = -\eta \cdot \delta_h[h] + \alpha \cdot \Delta \theta_h[h]$$

(2-17)

步驟七：更新各層間的加權值及偏權值。

說明：更新隱藏層與輸出層間的加權值及輸出層偏權值的方法如下：

for h=1 to Nhid

for j=1 to Nout

$$W_{hy}[h][j] = W_{hy}[h][j] + \Delta W_{hy}[h][j]$$

(2-18)

for j=1 to Nout

$$\theta_y[j] = \theta_y[j] + \Delta\theta_y[j]$$

(2-19)

同理，更新輸入層與隱藏層間的加權值及隱藏層偏權值的方法如下：

for i=1 to Ninp

for h=1 to Nhid

$$W_{xh}[i][h] = W_{xh}[i][h] + \Delta W_{xh}[i][h]$$

(2-20)

for h=1 to Nhid

$$\theta_h[h] = \theta_h[h] + \Delta\theta_h[h]$$

(2-21)

步驟八：重複步驟 3 至步驟 7，直到網路收斂。

說明：學習過程通常以一次一個訓練樣本進行，直到網路學習完所有的訓練樣本，稱為一個學習循環(learning circle)，我們可以讓網路重複學習個學習循環，直到網路收斂為止。為了測試網路是否收斂，我們定義下列誤差函數來表示網路的學習品質：

$$E = \left(\frac{1}{2}\right) \sum_j (T[j] - Y[j])^2$$

此式表示輸出層各個神經元的平方誤差和。因為在學習過程中，我們希望網路的推論輸出值 $Y[j]$ 與目標輸出值 $T[j]$ 越接近越好，所以(2-22)式的計算值應小於一個合理的範圍才行。

硬體設備

本系統所採用的 BS2-IC 微控制器[7]如圖 3

特色：

處理器速度：20 MHz

程序執行速度：~4000 PBASIC 的指令/秒。

RAM 大小：32 Bytes (6 I/O, 26 Variable)

EEPROM 的 (程序) 尺寸：KBytes; ~500 PBASIC 說明

I / O 腳位：16 + 2 專用串行

@ 5 VDC 的電流消耗：3mA 的運行，50 μ A 的睡眠

Source/Sink Current per I/O：20 毫安/二十五毫安

Source/Sink Current per unit:：40 毫安/ 50 毫安，每 8 個 I / O

腳位

PBASIC 的命令：42

封裝：24 引腳 DIP

規格:電源要求：5.5 至 15 伏直流電壓 (VIN) 或 5 伏直流電 (VDD)

通訊：串行 (9600 波特率編程)

尺寸：1.20 x 0.63 x 0.15 英寸 (30.0 x 16.0 x 3.81 毫米)

工作溫度：-40 至+185 ° F (-40 至+85°C)

所採用的 XBee 1mW Chip Antenna 通訊模組其特色為

特色：

- 3.3V (50mA)
- 資料傳送速率最大 250kbps
- 輸出 1mW (+0dBm)
- 工作範圍 300ft (100m)
- U.FL 天線接頭 1 FCC 認證
- 6 組 10-bit ADC input pins
- 8 組 IO pins , 128-bit 加密
- Local or over-air

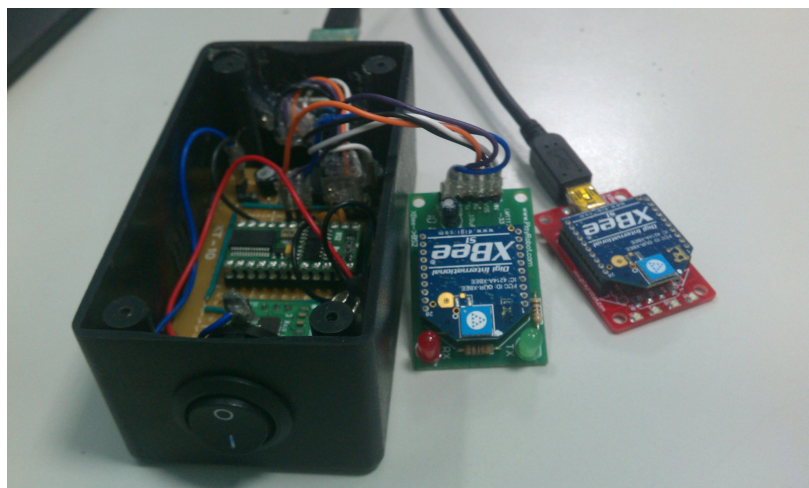


圖2: 收發模組

參、結果與討論

為了確認應用軟體的準確性，本研究以 $y = 2x + \sin(\pi x) + \sin(2\pi x)$ 做為測試問題。由圖3得知，類神經學習後所得結果與正解十分吻合。

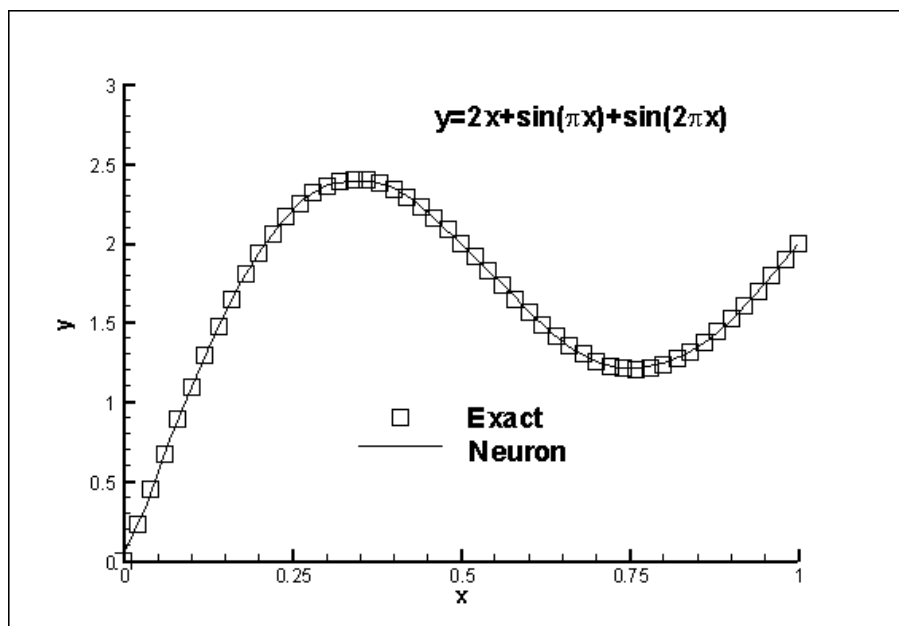


圖3：準確性驗證

圖4為系統操作介面，使用者可依實務需要設定神經元、學習循環次數及資料檔案名稱等參數。圖5為計算完成後之顯現結果。



圖4：操作介面



圖5: 計算結果與收斂情況

本團隊已完成跌倒偵測的研究能量，其成果如下：

1. 跌倒偵測系統符合功能設計需求，可做為居家或照護機構之人員管理使用。
2. 跌倒偵測系統符合可靠性、使用性及效率性的要求。
3. 未來將持續提升系統功能，結合GPS[8]及手機功能可及時將老年人活動訊息傳至管理平台，確保老年人的行動安全。

參考文獻

1. 內政部統計處，
http://www.moi.gov.tw/stat/news_content.aspx?sn=4943
2. Xinguo Yu， “Approaches and Principles of Fall Detection for Elderly and Patient”， 2008 10th IEEE Intl.Conf.on e-Health Networking， Application and Service
3. H. Nart-Charif and S.J McKema， “Activity summarization and fall detection in a supportive home environment” ， ICPR 2004
4. M.N. Nyan， Francis E.H Tay， M. Manimaran and K.H.W. Seah. ， “J. Phys: Conf Ser 34(2006) 1059-1067”
5. 資訊教育研究室， “Visual C# 2008”， 博碩文化， 2009
6. 領航數位科技， “Visual C# 2008 範例教學手冊”， 藍海文化， 2008
7. 普特企業有限公司； http://www.playrobot.com/home_index.htm
8. 劉仁筑、鄔智仁， “GPS 程式設計開發實戰”， 學貫圖書