SCUOLA DOTTORALE DI INGEGNERIA

SEZIONE DI INGEGNERIA DELL'ELETTRONICA BIOMEDICA, DELL'ELETTROMAGNETISMO E DELLE TELECOMUNICAZIONI

XXIV CICLO

Reti neurali artificiali innovative per l'inseguimento dello sguardo

Innovative Artificial Neural Networks for Eye Tracking

Ing. Massimo Gneo

Supervisor: Prof. Tommaso D'Alessio

Ph. D. Program Coordinator: Prof. Lucio Vegni

Sommario

I sistemi di inseguimento dello sguardo (*eye-gaze tracking*) stimano il punto osservato da un utilizzatore su di una superficie (es. il monitor collegato ad un personal computer).

Sono utilizzati in ambito *diagnostico*, per studiare le caratteristiche e le anormalità del sistema oculomotorio (es. in oftalmologia, neurologia, psicologia), e in applicazioni *interattive* dove il sistema costituisce il dispositivo periferico d'ingresso di un'interfaccia uomo-computer (es. per muovere il cursore sullo schermo quando il controllo attraverso il mouse non è possibile, come accade nel caso di sistemi di ausilio per pazienti gravemente disabili).

Se la testa dell'utilizzatore rimane ferma e la sua cornea è assunta sferica e rotante intorno al suo centro fisso, nelle immagini catturate da una o più telecamere, la pupilla segue l'occhio durante i suoi movimenti, mentre le riflessioni generate da una o più sorgenti di luce infrarossa sulla superficie esterna della cornea (i cosiddetti *glint*) possono essere assunte come punti fissi di riferimento.

La tecnologia più diffusa per i sistemi di eye-gaze tracking è la *pupil center corneal reflection*, che consiste nell'estrazione delle coordinate del centro della pupilla e dei glint dalle immagini dell'occhio, e nella successiva trasformazione (o *mapping*) di tali coordinate in quelle del punto osservato (Hutchinson et al., 1989).

Una delle caratteristiche peculiari di un sistema di eye-gaze tracking, quindi, è la c.d. *funzione di mapping* che effettua tale trasformazione.

* * *

Riguardo a tutte le possibili configurazioni di un sistema basato sulla pupil center corneal reflection, in termini di numero e posizione delle telecamere e delle sorgenti di luce all’infrarosso, alcuni importanti risultati teorici, di seguito brevemente riportati, hanno costituito la prima fonte d’ispirazione di questo lavoro di tesi (Guestrin and Eizenman, 2006; Villanueva and Cabeza, 2008):

- *1 telecamera, 1 sorgente IR*: con tale configurazione il punto osservato non può essere stimato se la posizione della testa dell’utilizzatore non rimane stazionaria o non viene stimata in altro modo,
- *1 telecamera, 2 sorgenti IR*: è la configurazione più semplice che consente la stima del punto osservato lasciando l’osservatore libero di muovere la testa; tale configurazione, inoltre, consente di ottenere un’accuratezza della stima del punto osservato di circa 1° in termini d’angolo visuale (è l’accuratezza generalmente accettata per i sistemi interattivi impiegati nelle interfacce uomo-computer);
- l’accuratezza può essere migliorata impiegando un numero maggiore di sorgenti IR,
- qualsiasi sia il numero delle telecamere e delle sorgenti IR utilizzate, è necessario effettuare una calibrazione del sistema.

Ciò considerato, in questa tesi si propone di utilizzare una sola telecamera e di incrementare il numero di sorgenti IR da due a tre così che la stima del punto osservato possa essere (teoricamente) effettuata anche con la testa in movimento, ottenendo, potenzialmente, un’accuratezza migliore di 1°.

Si è, pertanto, realizzato un sistema d’illuminazione in grado di provocare una particolare configurazione triangolare di tre glint proiettati sull’occhio dell’utilizzatore. L’informazione contenuta in tale configurazione, quindi, è stata opportunamente utilizzata per rivelare in maniera robusta le caratteristiche dell’occhio (es. consentendo di scartare artefatti nell’immagine), ed ha consentito, al contempo, di utilizzare dei sistemi d’illuminazione e di ripresa di complessità inferiore a quella dei sistemi di eye-gaze tracking studiati o commercializzati (evitando, in particolare, la necessità di sincronizzare i segnali di attivazione delle sorgenti di luce con quello di acquisizione dalla telecamera).

*   *   *

Lo studio della funzione di mapping da utilizzare per il sistema di eye-gaze tracking che utilizzi il sistema di illuminazione proposto ha fornito l'altro principio di base per questo lavoro di tesi.

La determinazione della funzione di mapping di un sistema di eye-gaze tracking può avvenire secondo due possibili approcci: quello basato sull'adozione di un modello del sistema e dell'occhio dell'utilizzatore (*model-based*), e l'approccio indipendente da modelli espliciti e basato sulla regressione (*regression-based* o, semplicemente, *model-independent*) (Hansen and Ji, 2010).

L'approccio *model-based* compie una stima diretta del punto osservato attraverso la derivazione in forma esplicita della funzione di mapping a partire da modelli geometrici (approssimati) relativi alla geometria dell'allestimento del sistema, ai componenti del sistema e all'occhio del particolare utilizzatore, caratterizzati, rispettivamente, da parametri fisici (es. lunghezza focale e posizione della telecamera) e fisiologici (es. il raggio della cornea approssimata come una sfera).

I sistemi di eye-gaze tracking di tipo model-based, quindi, soffrono degli svantaggi causati dall'adozione di un particolare modello esplicito, rigido e, inevitabilmente, approssimato, tra i quali:

- la scarsa flessibilità del sistema al variare dei parametri che descrivono: le caratteristiche fisiologiche degli occhi dei diversi utilizzatori, i particolari componenti utilizzati e il loro allestimento e assemblaggio,
- la limitazione dell'accuratezza ottenuta nella stima del punto osservato.

L'approccio *model-independent*, al contrario, stima la funzione di mapping impiegando tecniche di regressione, senza il bisogno, quindi, di assumere alcun particolare modello approssimato né per la fisiologia dell'occhio, né per l'allestimento del sistema, consentendo, potenzialmente, maggiore flessibilità e accuratezza.

Una delle tecniche di regressione più utilizzate è quella basata sulle reti di neuroni artificiali, più note come reti neurali, le quali, come ampiamente dimostrato, costituiscono dei regressori universali in grado di approssimare ogni funzione misurabile con qualsiasi desiderato livello di accuratezza (Cybenko, 1989; Hornik et al., 1989).

In questa tesi, pertanto, alla stregua di autori come (Baluja & Pomerleau, 1994; Piratla & Jayasumana, 2002; Zhu & Ji, 2004), adottando l'approccio *model-independent*, si è proposto di impiegare le reti neurali per apprendere la funzione di mapping del sistema di eye-gaze tracking proposto, basato sulla tecnologia pupil center corneal reflection, equipaggiato con il descritto sistema d'illuminazione a tre sorgenti IR.

Sebbene l'impiego di una funzione di mapping neurale possa ovviare, in teoria, agli svantaggi inerenti all'utilizzo di un sistema di eye-gaze tracking basato su modelli espliciti, alcuni pregiudizi accompagnano, storicamente, le applicazioni pratiche delle reti neurali tradizionali.

Oltre alle difficoltà correlate alla ricerca della migliore architettura dei collegamenti, altri inconvenienti dipendono dall'estensione illimitata del dominio delle funzioni sigmoidali che, tradizionalmente, realizzano le funzioni d'attivazione dei neuroni artificiali: la lentezza dell'apprendimento (Cybenko, 1989; Hornik, Stinchcombe, & White, 1989), la mancanza di significato fisico della rappresentazione realizzata dalla rete neurale durante l'addestramento, l'aggiornamento dei pesi delle connessioni (Rumelhart, Hinton, & Williams, 1986), l'interferenza negativa (Schaal & Atkeson, 1998) e la non realizzabilità del parallelismo nell'implementazione.

Questi aspetti negativi, ancora in gran parte irrisolti, possono pregiudicare l'esito dell'applicazione delle reti neurali per apprendere la funzione di mapping causando, in particolare, delle fasi di calibrazione eccessivamente durature (è durante tale fase iniziale, nella quale, generalmente, l'utilizzatore osserva dei punti di posizione nota proposti dal sistema, che avviene l'addestramento della rete).

Obiettivo di questa tesi, quindi, è stato lo studio e la ricerca di nuove architetture e schemi di addestramento in grado di superare i problemi correlati all'impiego di reti neurali per la stima della funzione di mapping di un sistema di eye-gaze tracking.

* * *

Allo scopo di superare gli svantaggi principali che, allo stato, affliggono ogni sistema di eye-gaze tracking studiato o posto in commercio, evitando l'assunzione di modelli espliciti, semplificando l'architettura e la complessità del sistema, aumentandone, al contempo, la robustezza e l'accuratezza, questa tesi ha approfondito, specificamente, i seguenti argomenti:

1. la realizzazione di un nuovo sistema di eye-gaze tracking indipendente da modelli e basato su reti neurali, equipaggiato con un sistema di illuminazione innovativo, con una struttura semplificata e capace di proiettare una opportuna configurazione di riflessioni sull'occhio dell'utilizzatore,

2. la predizione di serie temporali in tempo reale basata su reti neurali, con lo scopo di realizzare funzioni di mapping capaci di superare i problemi correlati ai falsi negativi nella rivelazione delle caratteristiche dell'occhio e al movimento della testa dell'utilizzatore del sistema,
3. l'introduzione di reti neurali basate sull'interconnessione di nuovi *campi ricettivi localizzati* costituiti da funzioni di attivazione di forma ellittica, allo scopo di ottenere, una funzione di mapping in grado di realizzare una rappresentazione fisicamente significativa del problema, con capacità di approssimazione simile a quella delle reti neurali tradizionali e velocità di apprendimento superiore,
4. un sistema di controllo di una sedia a rotelle elettrica che integra il sistema di eye-gaze tracking proposto e una interfaccia di tipo brain-computer tradizionale, capace di selezionare il comando di movimento desiderato attraverso lo sguardo dell'utilizzatore e di attivare, tale comando, attraverso il segnale elettroencefalografico.

Nel Capitolo 1 della tesi è descritto il contesto di riferimento, riassumendo lo stato dell'arte e le generalità sui sistemi di eye-gaze tracking e sulle reti neurali.

Riguardo al primo argomento, è stato realizzato il prototipo di un sistema di eye-gaze tracking basato su reti neurali ed equipaggiato con un nuovo sistema d'illuminazione costituito da tre sorgenti di luce in grado di generare una configurazione triangolare di tre glint sull'occhio dell'utilizzatore: tale informazione è sfruttata consentendo una maggiore robustezza della rivelazione delle caratteristiche dell'occhio (pupilla e glint) e di evitare, al contempo, la necessità di un circuito di sincronizzazione tra i sistemi d'illuminazione e di ripresa. L'impiego di reti neurali consente, inoltre, di stimare direttamente la funzione di mapping del sistema e di evitare l'assunzione di modelli espliciti, realizzando un sistema a geometria variabile, grazie al quale l'utilizzatore e i componenti del sistema possono essere liberamente sostituiti e assemblati. La fattibilità del sistema proposto è stata provata nel Capitolo 2, dove sono stati riportati i risultati relativi ai test effettuati durante più sessioni di funzionamento in tempo reale.

La robustezza del sistema proposto è stata anche provata in maniera dettagliata nel Capitolo 3 dove è stata effettuata la valutazione dell'accuratezza raggiunta impiegando dati provenienti da sessioni reali di funzionamento realizzate da: i) diversi utilizzatori; ii) diversi allestimenti in termine di posizione della telecamera e del sistema d'illuminazione; iii) diversi protocolli di test basati sull'osservazione sia dei punti su di una griglia rettangolare di calibrazione, sia dei punti di una griglia di test, intermedi ai precedenti. L'accuratezza ottenuta è stata non superiore a 0.49°, 0.41° e 0.62° per gli errori di stima del punto osservato, rispettivamente alla direzione orizzontale, verticale e radiale. Il sistema proposto, pertanto, ha dimostrato di poter raggiungere prestazioni migliori dei sistemi di eye-gaze tracking progettati per l'interazione uomo-computer i quali, sebbene equipaggiati con hardware superiore, raggiungono accuratezze aventi valori tipici compresi tra 0.6° e 1°.

Con riferimento al secondo argomento, allo scopo di superare i problemi correlati ai falsi negativi nella rivelazione delle caratteristiche dell'occhio e al movimento della testa dell'utilizzatore di un sistema di eye-gaze tracking, nel Capitolo 4 sono stati proposti degli schemi di predizione di serie temporali basati sulle reti neurali utilizzate per il calcolo delle funzioni di mapping del sistema di eye-gaze tracking descritto nel Capitolo 2 e nel Capitolo 3.

Tali schemi di predizione sono stati applicati con successo al riconoscimento di gesti dell'arto superiore, considerando le serie temporali ottenute dalle uscite di due accelerometri posizionati sul braccio e sull'avambraccio. Gli errori di predizione sono stati utilizzati sia per l'addestramento delle reti neurali di predizione, sia per la stima di una misura della verosimiglianza dell'occorrenza di ogni specifico gesto.

In modo conforme all'approccio indipendente da modelli adottato, non sono state fatte assunzioni a priori né elaborazioni preliminari delle serie temporali oggetto di predizione. Sui quattro gesti considerati, il metodo proposto ha raggiunto una percentuale di riconoscimenti corretti superiore all'83%. Ciò ha incoraggiato l'integrazione dello schema di predizione proposto nella funzione di mapping del sistema di eye-gaze tracking descritto nel Capitolo 2 e nel Capitolo 3.

Il supporto illimitato delle funzioni di attivazione sigmoidali utilizzate nelle reti neurali multistrato tradizionali provoca lentezza nell'apprendimento, assenza di significato fisico della rappresentazione costituita dalla rete addestrata, interferenza negativa tra i neuroni per le diverse configurazioni degli ingressi della rete. Ciò può pregiudicare l'impiego di reti neurali nei sistemi di eye-gaze tracking causando, in particolare, calibrazioni eccessivamente onerose.

I campi ricettivi localizzati costituiscono funzioni di attivazione aventi un supporto limitato. Le reti che interconnettono tali "neuroni", offrono, potenzialmente, potere di approssimazione analogo a quello esibito dalle reti neurali multistrato convenzionali, con maggior velocità nell'apprendimento e fisica significatività delle rappresentazioni ottenute (Powell, 1987; Park & Sandberg, 1991; Park & Sandberg, 1993). Tali reti, tuttavia, hanno spesso dimensioni eccessive, e/o prestazioni peggiori delle reti neurali convenzionali a causa della determinazione non supervisionata della posizione e del fattore di forma dei campi ricettivi, che non impiega, quindi, tutta l'informazione disponibile, e dell'impiego di campi ricettivi simmetrici, tutti di forma simmetrica prefissata e identici tra loro (l'addestramento di tali reti, quindi, determina le sole altezze di ciascun campo ricettivo).

Con riferimento al terzo argomento, quindi, nel Capitolo 5 sono stati introdotte delle reti di nuovi campi ricettivi localizzati, chiamati *quadratic exponential elliptical neurons* (QuEEN), che possono essere ricondotte a opportune reti neurali multistrato consentendo di l'applicazione dell'algoritmo standard di retro propagazione dell'errore (*backpropagation*). Ogni campo QuEEN, quindi, può essere posizionato e sagomato contestualmente, durante un addestramento supervisionato. Le simulazioni numeriche effettuate, infatti, hanno dimostrato che le reti di QuEEN sono in grado di esibire un potere di approssimazione simile a quello delle reti neurali multistrato convenzionali, con un tempo di addestramento inferiore.

Con riferimento all'ultimo argomento, grazie all'indipendenza da modelli e alla possibilità di posizionare liberamente l'utilizzatore e i componenti impiegati, nel Capitolo 6 è stata analizzata l'applicazione del sistema di eye-gaze tracking proposto al controllo di sedie a rotelle elettriche.

Tutti gli analoghi sistemi di controllo richiedono un'interfaccia grafica che l'utilizzatore deve osservare per selezionare e confermare il comando di movimento della sedia a rotelle. Tale modalità di controllo, tuttavia, consente una guida piuttosto innaturale, causa una parziale ostruzione della vista, e rende necessario l'impiego di segnali di controllo indipendenti dallo sguardo.

Grazie alla flessibilità del sistema di eye-gaze tracking proposto, è stato proposto di integrare il controllo attraverso lo sguardo con una *brain-computer interface* tradizionale. L'utilizzatore, quindi, potrà selezionare il movimento desiderato con gli occhi e confermare l'attivazione del movimento attraverso il segnale elettroencefalografico. Il sistema integrato proposto, quindi, consente di realizzare un controllo sicuro della sedia a rotelle, non ostacolando la vista con monitor e/o display

*   *   *

Alcuni importanti problemi ancora affliggono ogni sistema di eye-gaze tracking studiato o commercializzato e molto deve ancora essere fatto per superarli: i risultati ottenuti in questa tesi potrebbero offrire, auspicabilmente, utili spunti e suggerimenti per ridurre gli svantaggi correlati alla scarsità della robustezza e dell'accuratezza, principalmente dovuti all'approssimazione implicitamente contenuta nei modelli espliciti adottati.

Verso tali obiettivi, a partire dai risultati ottenuti, merita ulteriore approfondimento l'applicazione ai sistemi di eye-gaze tracking sia degli schemi di apprendimento, sia delle architetture neurali proposte in questa tesi.

Ringraziamenti

Nella mia tesi di laurea non riportai ringraziamenti. Per pudore e, forse, con un pizzico di anticonformismo. Non ricordo bene. Sedici anni dopo sono meno pudico e, almeno per certe cose, ho imparato ad apprezzare un pizzico di conformismo.

Ringrazio per prima mia moglie Dorota, che ha sostenuto la scelta, non facile per me, di frequentare questo dottorato, e le nostre due piccole creature Victor e Valerio: a loro tre ho sottratto non poco tempo. Non hanno capito tutto quello che facevo, ma hanno capito che per me era importante.

Molti anni fa il Prof. Tommaso D'Alessio telefonò a casa di uno studente per avvisarlo che l'esame di Elaborazione dei Dati e dei Segnali Biomedici era rinviato e per concordare la nuova data. In quegli anni, nell'ateneo più grande d'Europa, quella era un'eccezione che non ho saputo dimenticare. Durante questo dottorato il mio Docente Guida, il Prof. D'Alessio, ha messo al mio servizio la sua vasta competenza scientifica, ascoltato le mie fantasiose ipotesi di lavoro durante *brainstorm* indimenticabili, non lesinando, in alcuni momenti di difficoltà personale, parole e abbracci d'incoraggiamento. Non dimenticherò nemmeno tutto questo.

Oltre al Prof. D'Alessio, ringrazio anche la Prof.ssa Silvia Conforto (entrambi non usiamo sprecare parole, ma ci siamo sempre capiti e compresi, anche piuttosto oltre le poche parole) e il Prof. Maurizio Schmid (sempre paziente, disponibile e sensibile: è riuscito a non farmi pesare le sue sovrastanti doti umane e scientifiche).

Ricordo, in stretto ordine alfabetico, tutti gli altri amici del Biolab: Daniele Bibbo, Ivan Bernabucci, Margherita Castronovo, Baldassarre Baldo D'Elia, Cristiano De Marchis, Michela Goffredo, Rossana Muscillo, Giacomo Severini, Diego Torricelli e Raffaella Spica della Segreteria didattica, che non s'è mai stancata di darmi consigli e di supportarmi nelle pratiche amministrative: senza di loro il lavoro sarebbe stato più difficile le giornate trascorse meno piacevoli.

Non posso trascurare di ringraziare il Dr. Davide Carbone: senza l'aiuto che mi ha dato e la passione profusa nel farlo, questo lavoro (forse) sarebbe stato, in ogni caso, portato a termine, ma con maggiori difficoltà e senza l'entusiasmo e l'eccitazione che, invece, lo hanno accompagnato.

Come non ricordare, infine, Paolo Carbone, Maestro di vita, senza il quale questo lavoro sarebbe stato, in ogni caso, portato a termine, ma senza aver ricevuto in dono le perle di saggezza (e le risate) che, invece, mi sono state disinteressatamente elargite.

Roma, 17 febbraio 2012 Massimo Gneo

"Ogni libro contiene la sua pagina peggiore e la sua migliore, e quindi anche il mio. Tutte e due sono per me imbarazzanti. Con la migliore temo di avere persuaso qualcuno del mio pensiero, e me ne pesa la responsabilità, con la peggiore temo di avere dissuaso qualcuno da un'idea che forse era giusta."
(Giuseppe Sermonti, L'anima scientifica)

"Tutti i fenomeni naturali sono in definitiva interconnessi, e per spiegare uno qualsiasi di essi dobbiamo comprendere tutti gli altri, il che, ovviamente, è impossibile.
I grandi successi della scienza sono dovuti alla possibilità di introdurre approssimazioni. [...] Così è possibile spiegare un gran numero di fenomeni a partire da alcuni di essi, e di conseguenza si possono capire diversi aspetti della natura in modo approssimativo senza dover comprendere tutto quanto in una volta sola.
Questo è il metodo scientifico; tutte le teorie e i modelli scientifici sono approssimazioni della vera natura delle cose, ma l'errore che si introduce con l'approssimazione è spesso sufficientemente piccolo da giustificare questo modo di procedere."
(Fritiof Capra, Il Tao della fisica)

"Di più, nin zò!"
(Fabrizio Maturani, in arte Martufello)

Keywords

Eye Gaze Tracking

Pupil Center Corneal Reflection

Artificial Neural networks

Radial Basis Functions

Localized Receptive Fields

Self-Organizing Networks

Time Series Prediction

ABSTRACT

Eye-gaze tracking systems estimate the point of gaze of an user.

According to the consolidated pupil center corneal reflection technique, the coordinates of the pupil and outer corneal reflections of the user's eye images are mapped onto the coordinates of her/his gaze on the observed surface (e.g. a computer display).

Contrarily to the model-based approach, model-independent method estimates the mapping function by means of regression techniques with no need of any specific model assumption and approximation either for the user's eye physiology or the system initial setup.

This thesis describes novel architectures and learning schemas of artificial neural networks conceived to regress the mapping function of eye-gaze tracking systems.

A new neural eye-gaze tracking system admitting a free geometry positioning for the user and the system components is conceived and built.

The accuracy and the robustness of the proposed system are tested and proved.

In order to overcome the problems due to failures in eye features detection and head motion, a time series prediction neural scheme is also proposed and verified on real data.

With the objective of reduce the duration of the system calibration, a new artificial neuron implementing elliptical localized receptive fields is proposed and tested, obtaining comparable regression power and faster learning than conventional multilayer neural networks.

TABLE OF CONTENTS

Introduction

"*[...eye tracking output] is estimation of the projected Point Of Regard*[*] *(POR) of the viewer, i.e., the (x,y) coordinates of the user's gaze on the computer display. [...] why is eye tracking important? Simply put, we move our eyes to bring a particular portion of the visible field of view into high resolution [...] most often we also divert our attention to that point so that we can focus our concentration [...] on the object or region of interest. Thus, we may presume that if we can track someone's eye movements, we can follow along the path of attention deployed by the observer. This may give us some insight into what the observer found interesting [...]*" (Duchowsky, 2007).

Eye-Gaze Tracking Systems (EGTSs) are also used in *diagnostic* applications to study oculomotor characteristics and abnormalities (e.g. in ophthalmology, neurology, psychology), whereas in *interactive* applications EGTSs are proposed as input devices for human computer interfaces (HCI), e.g. to move a cursor on the screen when mouse control is not possible, such as in the case of assistive devices for people suffering from locked-in syndrome.

How the most diffused EGTSs work? If the user's head remains still and the cornea rotates around its fixed centre, the pupil follows the eye in the images captured from one or more cameras, whereas the outer corneal reflections generated by one or more infrared (IR) light sources, i.e. *glints*, can be assumed as fixed reference points. According to the well known pupil centre corneal reflection method (PCCR), the system *mapping function* maps glints and pupil centers in the eye image onto the POG coordinates (Hutchinson et al., 1989).

[*] Point Of Regard (POR) or Point Of Gaze (POG).

Some important results were found about PCCR-based EGTSs covering all the possible system configurations in terms of number and positioning of IR light sources and cameras (Guestrin and Eizenman, 2006; Villanueva and Cabeza, 2008):

- *1 camera, 1 IR source*: the POG cannot be estimated unless the head is stationary or the head position is estimated by some other means,
- *1 camera, 2 IR sources*: is the simplest configuration that allows estimating the POG letting the head free,
- regardless of *how many* cameras or IR sources are used, system calibration is necessary,
- *1 camera, 2 IR sources*: is sufficient (about 1° of accuracy), whereas the use of more IR sources and calibration points increase the accuracy.

Considered the above results, we firstly propose to use one camera and increase the number of IR lights from two to three so that: the theoretical POG estimation may be performed even when the head moves, the potential system accuracy is lower than 1°, and the triangular pattern of glints projected on the user's eye can be exploited to allow convenient and robust eye feature detection, whereas the illuminating system complexity is kept low.

The main approaches for implementing EGTSs are the *model-based* and the *regression-based* – which we prefer to refer to as the *model-independent* – methods (Hansen and Ji, 2010).

The *model-based* approach directly estimates the POG by using an explicit implementation of the mapping function derived from approximated geometric models characterized by physiological and physical parameters respectively related to the user's eye (e.g. radius of the cornea approximated as a sphere), and to the geometry of the system.

The alternative *model-independent* approach estimates the mapping function by means of regression techniques with no need of any specific model assumption and approximation either for the user's eye physiology or the system initial setup admitting a free geometry positioning for the user and the system components.

As artificial neural networks (ANNs) are shown to be universally able to approximate any measurable function to any desired degree of accuracy (Cybenko, 1989; Hornik et al., 1989), we propose to use ANNs to learn the mapping function of a new model-independent EGTS based on PCCR and equipped with an innovative and simplified illuminating system, as other authors did (e.g. Baluja & Pomerleau, 1994; Piratla & Jayasumana, 2002; Zhu & Ji, 2004).

"*Although there exist many different approaches [...] for finding the optimal architecture of an artificial neural network, these methods are usually quite complex in nature and are difficult to implement. [...] hence the design of an ANN is more of an art than a science.*" (Zhang & Patuwo, 1998).

If a neural mapping function would theoretically overcome problems and constraints related to model-based EGTSs, some age-old drawbacks have raised some skepticisms on practical application of traditional ANNs. In addition to the difficulties in finding the optimal architecture of ANNs, we highlight those due to the infinite support of sigmoidal activations: the slow learning rate (Cybenko, 1989; Hornik, Stinchcombe, & White, 1989), the lack of physical meaning of the representation built during the training (Rumelhart, Hinton, & Williams, 1986), the negative interference, and the unfeasibility of parallel implementation.

Those open issues may prevent the useful application of ANN on EGTS giving, in particular, slow calibrations.

Therefore, the aim of this thesis is to devise innovative neural architectures to overcome the drawbacks of traditional ANNs in order to implement the mapping function of the mentioned new PCCR-based EGTS that use a convenient illuminating system and keeps the advantages of model-independent approach.

Namely, the main contribution of this thesis regards the following open issues related to EGTSs and traditional ANNs:

- proposing novel ANN architectures to implement a model-independent mapping function,
- keeping the same (or better) regression power of traditional ANNs, overcoming the drawbacks related to infinite support of sigmoidal activations (slow convergence rate, lack of physical meaning, etc.)
- avoiding any specific model assumption and approximation either for the user's eye physiology or the EGTS initial setup,
- admitting a free geometry positioning for the user and the system components,
- keeping low the complexity of the EGTS illuminating system,
- allowing a theoretical accuracy better than 1°.

Thesis Overview

Chapter 1 gathers some information on both ANNs and POG estimation and summarizes the state of the art in these fields highlighting, in particular, the main drawbacks still plaguing practically each EGTS studied or sold. The rationales and the guidelines of the research, aimed to give contribution on those open issues are then briefly introduced.

Chapter 2 presents a first study about the proposed PCCR-based EGTS which mapping function is given by the most known ANNs, the Multilayer Neural Networks (MNN). A prototype of the EGTS is built and successfully tested during several sessions of real operation, so proving the feasibility of the proposed approach.

The issue of avoiding any specific model assumption and approximation either for the user's eye physiology or the system initial setup is extensively analyzed in Chapter 3. The free geometry positioning for the user and the system components is tested and the robustness of the proposed EGTS is proven.

Given the feasibility of EGTS with neural based mapping function, the research on new ANN architectures and learning schemes aimed to estimate the POG was then encouraged.

In order to overcome the problems due to failures in eye features detection and head motion, Chapter 4 specifically deals with real-time time series prediction based on ANNs.

Traditional multilayer neural networks (MNNs) may exhibit slow learning rate, lack of physical meaning, and negative interference. This may prevent the useful application of ANN on EGTS giving, in particular, slow calibrations. In Chapter 5 are then presented networks of new artificial neurons showing comparable regression power and faster learning than MNNs.

Those properties allow to investigate new fields of applications of EGTSs such as the control of electric-powered wheelchair.

In Chapter 6 is presented a high level scheme of a system integrating the proposed EGTS with a brain-computer interface so to obtain a safer obstruction-free eye- and brain guided electric-powered wheelchair.

The last chapter draws the general conclusions of the thesis work.

Chapter 1

Background and Rationales

ABSTRACT

Eye gaze tracking systems estimate the point of gaze (POG) of an user. The coordinates of the pupil and outer corneal reflection generate by IR light sources in the eye image captured from one or cameras are generally mapped onto the POG coordinates.

Model-based approaches explicitly derive the mapping function from approximated geometric models characterized by parameters related to the user's eye and to the geometry of the system setup. The alternative model-independent approaches estimate the mapping function by means of regression techniques with no need of any specific model assumption and approximation, thus admitting a free geometry positioning for the user and the system components.

Although Artificial Neural Networks (ANNs) have been used to estimate mapping function in model-independent EGTSs, some age-old open issues may prevent this kind of application. In particular the slow convergence may give slow calibrations. Localized receptive field (LRF) networks have promised similar regression power and faster learning than conventional multilayer neural networks (MNNs).

EGTSs, MNN, and LRF networks are then briefly reviewed.

1.1. Basics on eye gaze tracking systems

Eye-gaze tracking systems (EGTSs) estimate the Point Of Gaze (POG) of an user.

Applications of EGTSs can be classified as *diagnostic*, where the user's visual and attentional processes are quantified, or *interactive* where the user inter-acts with the EGTS (Duchowski, 2002): in the first case, the obtained data are used to study oculomotor characteristics and abnormalities (e.g. in ophthalmology, neurology, psychology); in the second scenario, EGTSs are proposed as input devices for human computer interfaces (HCIs), e.g. to move a cursor on the screen when mouse control is not possible, such as in the case of assistive devices for people suffering from locked-in syndrome.

Sought-after requirements in EGTSs include minimal intrusiveness and obstruction, reduced calibration phase, allowing for free head movements, keeping high the accuracy and the setup flexibility, and maintaining low the cost.

Many traditional solutions for EGTSs are intrusive, as they require a physical contact with the user (e.g. contact lenses, reflective dots placed directly onto the eye, electrodes fixed around the eye, bitten and/or head mounted devices).

The EGTSs based on *video-oculograpy* (VOG) (i.e. *video-based* EGTSs), non-intrusively estimate the POG from the information given by the eye images captured from one or more cameras (Morimoto & Mimica, 2005; Hansen & Ji, 2010). Because of its minimal obtrusiveness, relatively easy set-up and dependence on optical and electronic imaging devices, VOG has become the most popular eye-tracking technique. VOG systems based on visible light are called *passive light* (Torricelli et al., 2008), whereas the ones using infrared (IR) are called *active light*. Nowadays, the latter are the most used thanks to numerous advantages: very little subject awareness (users are neither distracted nor disturbed by IR); strong iris reflectance in the near-IR, which grants well-contrasted images irrespectively of iris color, thus easing the pupil detection; low cost, since IR light can be provided by cheap IR light-emitting diodes (ILEDs) and captured by commercial charge-coupled device (CCDs) cameras.

1.2. Corneal reflection technology

The *pupil centre corneal reflection* (PCCR) is the *active light* eye-gaze tracking method par excellence (Hutchinson et al., 1989). If the user's head remains still and the cornea rotates around its fixed centre, the pupil follows the eye in the captured images, whereas the outer corneal reflection generated by an IR light source, i.e. *glint*, can be assumed as a fixed reference point (Figure 1.1, left). The POG can be thus estimated from the pupil-glint vector.

Both the glint and pupil centre locations can be easily extracted from the images captured by a camera under IR light. The glint appears in the IR band as a small intense spot whereas the pupil can be captured thanks to two distinct effects generated by IR: the *bright eye* (Figure 1.1 right) if the IR light source is close to the optical axis (*on axis*), and the *dark pupil* (Figure 1.1, left) if the IR light source is placed away from the camera (*off axis*) (Ebisawa, 1998; Morimoto et al., 2000; Zhu & Ji, 2004).

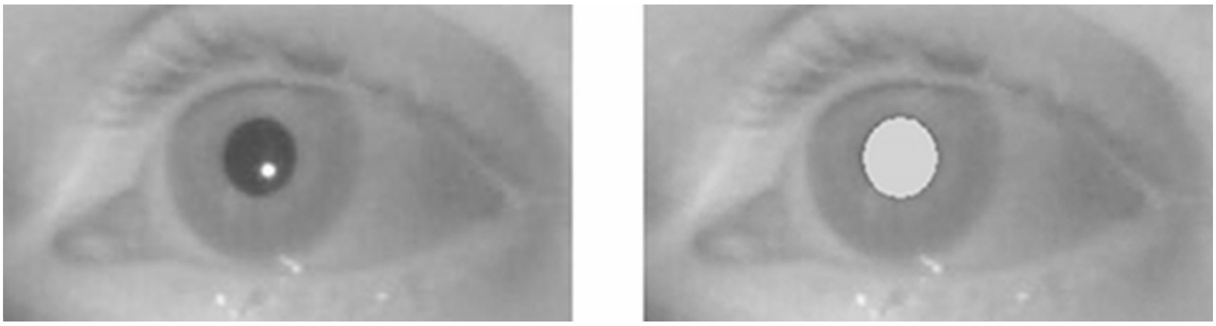


FIGURE 1.1 *Dark pupil and glint (left), bright eye (right)*

In a generic PCCR-based EGTS (Figure 1.2) the *mapping function* maps glints and pupil centers in the image onto the POG coordinates. The mapping function is the main typifying characteristic of an EGTS, and is determined through a *calibration* phase during which the user is asked to gaze at a proper set of known points on the observed surface.

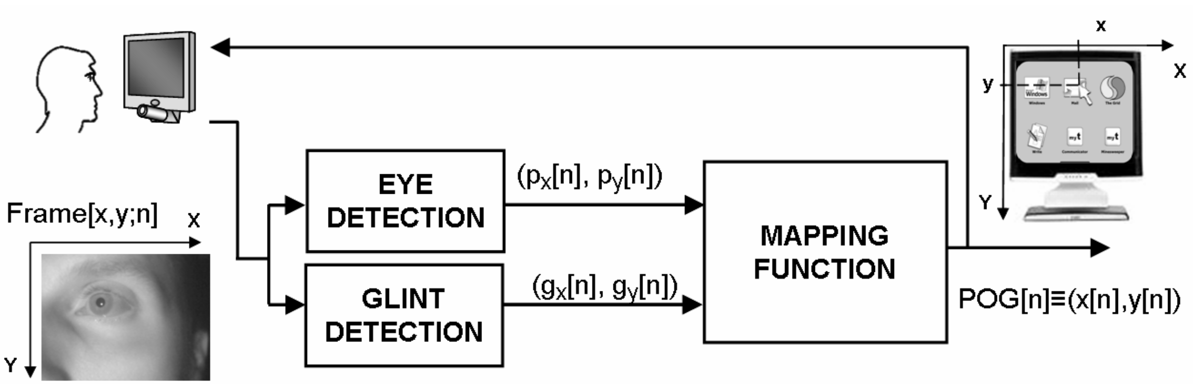


FIGURE 1.2 *Generic scheme of an EGTS based on pupil centre corneal reflection*

The most popular approach for implementing EGTSs is the so called *feature-based* method relating the POG to local eye features such as the pupil and glints for PCCR, the parameters of the system components setup, and the parameters of the eye physiology. Feature-based methods include the *model-based* and the *regression-based* – which we prefer to refer to as the *model-independent* – approaches (Hansen & Ji, 2010).

The former approach directly estimates the POG by using an explicit implementation of the mapping function derived from geometric models. These models are characterized by physiological and physical parameters related respectively to the user's eye (e.g. radius of the cornea approximated as a sphere), and to the geometry of the system setup (basically the camera features and the positions of the light sources, monitor, and user's eye corneal centre).

The *model-independent* approach estimates the mapping function by means of regression techniques, using either parametric (e.g. polynomial) or non-parametric forms (e.g. neural networks), whose coefficients have no physiological or physical meaning.

Both model-based and model-independent methods need a calibration phase to determine the model parameters and the regression coefficients respectively, when the user is asked to gaze at a set of predefined known points on the screen. The analogy between the two approaches is obviously maintained when the conditions move away from the calibration situations and the accuracy quickly decays if the user POG is far from the calibration points.

Model-based methods may simplify – but not avoid at all – the calibration to evaluate the parameters of the model (Guestrin & Eizenman, 2006; Villanueva & Cabeza, 2008).

1.3. Model-based Methods

We now proceed to argue about the main characteristics and drawbacks of model-based EGTSs.

While the camera and the geometric parameters may be directly measured or estimated once and for all during the first system setup, the physiological parameters of the eye are difficult to measure and are affected by large inter-subjects variability. This makes a calibration phase unavoidable. It was indeed shown that even for an EGTS where the simplified corneal spherical model is adopted and both the camera parameters and the system geometry are perfectly known, the POG determination still needs several physiological parameters, including: the ray of the corneal curvature, the distance between the pupil and the corneal centre, the combined index of refraction of the aqueous humor and cornea, and the angular offset between the optical and visual axes (Guestrin & Eizenman, 2006; Villanueva & Cabeza, 2008).[*]

Moreover, it has to be outlined that the increase of the measurements during the setup (needed for the parameters estimation), decreases the complexity of the calibration procedure, but at the same time it decreases also the flexibility of the system.

The values of the parameters provided by the estimation procedure are valid if the EGTS components are left in the same configuration used for the estimation. Any change in the configuration will affect the mapping function and cause an incorrect POG estimation.

The approximation process, that is typical of any model-based method, often makes the model even oversimplified. This is what happens with the ubiquitous corneal spherical model: it is particularly unsuitable for the outer regions of the cornea, where the corneal surface bends towards the sclera (Droege et al., 2007), leading to high inaccuracy when the user moves the eye to the extremities of the screen and the glint falls onto a non spherical surface. The oversimplification of the model has been reported as one of the main sources of the POG estimation errors.

To sum up, the adoption of whatever model-based approach involves the following main drawbacks:

[*] The *optical axis*, i.e. the eye symmetrical axis, is the line joining the pupil and cornea centre; the *visual axis*, i.e. the gaze direction, is the line joining the POG and the fovea, the highest acuity area of the retina, slightly displaced from the back pole of the eyeball.

1. the accuracy is limited by the approximation inherent in each model,
2. the initial setup of the system is relatively complex (the model parameters have to be accurately measured),
3. the system is rigidly bound to the initial setup (once measured, the model parameters must be kept fixed).

Moreover, regardless of the model complexity, the calibration can be only simplified but not avoided at all.

1.4. Neural model independent methods

The main difficulty with POG estimation is due to the inherent high complexity and nonlinearity of the mapping function, that is particularly severe with large pupil-glint vectors. That difficulty was already faced by model-independent methods by using classical polynomial regression (Morimoto et al., 2000) but, as with model-based approaches, the performance quickly decays when POG falls far from the calibration points.

As artificial neural networks (ANNs) – and particularly standard Multilayer Neural Networks (MNNs) – are shown to be universally able to approximate any measurable function to any desired degree of accuracy (Cybenko, 1989; Hornik et al., 1989; Scarselli & Tsoi, 1998), we propose to use MNNs as a multivariate non-linear mapping (Bishop, 1994) to learn the mapping function of a new PCCR-based EGTS.

ANNs are a biologically inspired computational paradigm using many simple elaboration units (*neurons*) highly interconnected. A set of significant inputs and corresponding desired output couples (*training set*) is used to *train* the ANNs connections strengths (*weights*) minimizing the error between the desired and actual outputs.

The *generalization* power of ANNs is related with the ability to correctly predict the output value for inputs not contained in the training set. The level of generalization reached at the end of the training is related to both the content of the training set and the complexity of the ANN in terms of the number of neurons and their interconnections:

- regarding the training set, the better the (input, output) domains are sampled, the higher is the generalization ability of an ANN,
- as regards the ANNs complexity, oversimplified ANNs can be unable to identify complicated behaviors (*underfitting*), whereas too complex ANNs may learn the noise affecting the training set data (*overfitting*), becoming unable to

correctly behave in conditions far from the contents of the training set.

We speculate in the following about how the appropriate use of MNNs allows overcoming *both* the drawbacks of the model-based EGTSs *and* the potential reasons of those failures that sometimes gave ANNs an undeserved not so good reputation.

The approximation inherent in whatever adopted model may be avoided as ANNs may in principle approximate with the desired degree of accuracy whatever complex EGTS mapping function. Moreover, thanks to their learn-by-examples ability, ANNs may learn any mapping function whatever is the configuration given to the system during the first setup. Therefore, the direct measurement or estimation of the model parameters during the system setup may be also bypassed and implicitly included in the learning of the function mapping, simplifying the setup process itself.

In addition, granted an opportune training set and the right complexity of the ANN, the generalization power of ANNs allows overcoming the problem generally afflicting both model-based and model-independent EGTSs, regarding the accuracy decay when the user's POG falls on points far from the calibration ones. Uniform accuracy all over the screen may be thus assured.

Above considerations stand if the theoretical behavior of ANNs is hypothesized. Although ANNs have been already used as EGTSs mapping functions (a brief review about (Baluja & Pomerleau, 1994), (Piratla & Jayasumana, 2002), and (Zhu & Ji, 2004) will be given in Section 3.2.4) with large training sets of eye images, the achieved POG estimation accuracy was not as good as for other techniques. Proven that MNNs are universal and arbitrarily accurate approximating tools, any failure in their application may arise from one or more of the following reasons (Hornik et al., 1989):

1. lack of deterministic input-output mapping,
2. unmet learning and/or training,
3. improper complexity of the ANN with respect to the problem.

The above three adverse situations can be avoided if MNNs are appropriately exploited as mapping functions of a PCCR-based EGTS.

The first topic can be excluded, as the real problem related to the mapping function of a PCCR-based EGTS is not to prove its existence but rather its inherent complexity.

As regards the inadequacy of learning and training, we believe that the EGTS calibration phase is a very good source of data to build an ideal training set. When the user is asked to gaze at a known point, the point coordinates provide the desired outputs, whereas the correspondingly captured eye features provide the related inputs. The training set is so built in correspondence of all the points on the calibration grid, and the codomain of the mapping function corresponds to all the coordinates of the monitor pixels.

This output space shows the following interesting properties:

- it is finite dimensional (2-D),
- it is bounded with exactly fixed boundaries (the monitor frame), and it has finite cardinality.

The codomain of an EGTS mapping function can be thus easily sampled giving a training set that can be arbitrarily made large and uniformly representative of the mapping itself. This is a crucial topic as it is well recognized that overfitting is very dangerous and the best way to overcome it is to build large training sets (Zhang, 1998; Crone, 2005).

The last topic, regarding the complexity of MNNs, implies the selection of the best architecture in terms of number of hidden layers, size of each layer, and interconnections. It is well recognized that this problem is so task-dependent that none of the known methods can be assumed as superior to the others (Crone, 2005). Though a heuristic trial-and-error approach is often used, especially about the hidden layers, some general rules may be given about the number of input and output neurons. As one of the golden rules of thumb is that the parsimonious architectures have the best performance and the highest generalization capability (Zhang, 1998), we believe that the ANNs so far used for POG estimation are too expensive in terms of both complexity and computational cost. An appropriate preliminary phase of eye features extraction on the images should be performed to maximize the compression of the information and minimize its loss, so that the number of ANN inputs is minimized too. This is the approach that has been sought in the EGTS here proposed.

Since ANNs are shown able to learn and approximate mappings from examples to any desired degree of accuracy and we believe that the POG determination is a well posed task for ANNs, we propose to adopt a model-independent approach based on ANNs to overcome the drawbacks of the model-based methods.

While a 1° accuracy is an agreed bound for the specifications of EGTSs designed as input devices for HCIs, we aim at a lower bound of 0.6° accuracy, coming from the physiological evidence that in the fovea, the highest acuity retinal area ranges from 0.6° to 1° (Guestrin & Eizenman, 2006).

1.5. Artificial Neural Networks

Artificial neural networks (ANNs) belong to a neurologically inspired computational paradigm that uses many simple elaboration units (neurons) highly interconnected.

When in a supervised scenario, a set of significant inputs and corresponding target output pairs (*training set*) is used to *train* the ANNs connections strengths (*weights*) minimizing the distance between the target and the actual outputs. According to the neurological long-term potentiation principle – the efficacy of synapses change as a result of experience providing both memory and learning to the brain – the training reinforces or depresses the connections giving the ANN the capability to learn the knowledge and behavior contained in the training set.

The *generalization* power of ANNs is related with the ability to correctly predict the output values for inputs not contained in the training set. Learning and generalization are among the most useful attributes of ANNs (Widrow & Lehr, 1990).

When ANNs are applied to a regression task, learning corresponds to finding a surface on the input space that gives the best fit to the training data following some optimum criterion, whereas generalization means interpolation between (and possibly extrapolation outside the range of)the sample data points along the regressing surface built during the training.

Multilayer neural networks (MNNs) are ANNs whose units are disposed in fully connected layers: each unit of each layer receives as inputs the outputs of every unit of the preceding layer. MNNs belong to the larger class of feedforward neural networks (FNNs), where the data processing flows from the input nodes towards the output, and the related graphs have no cycles.

Each unit of a MNN mimics the *all-or-none* behavior of biological neurons, which give a complete (and limited) response if stimulated above a certain activation threshold or, otherwise, give no response at all. This behavior is traditionally modeled by *squashing* sigmoidal functions quickly saturating as input move away from a threshold towards negative and positive values (the *logistic* monotonically grows assuming values in [0,1], whereas the hyperbolic tangent similarly ranges in [-1,+1]).

Since the single-hidden layer MNN class – the simplest nontrivial class of FNNs – with sigmoidal hidden units gives universal arbitrarily good approximators (Cybenko, 1989; Hornik et al., 1989), in this work we only consider one-hidden layer one-output MNNs (the extension to multidimensional output is obvious) whose graph is depicted in Figure 1.3, where each arch is a trainable weight and the sigmoidal unit thresholds are given by the weights of a trivial bias unit *b* having output fixed to 1. We refer to this class of ANNs as conventional MNNs.

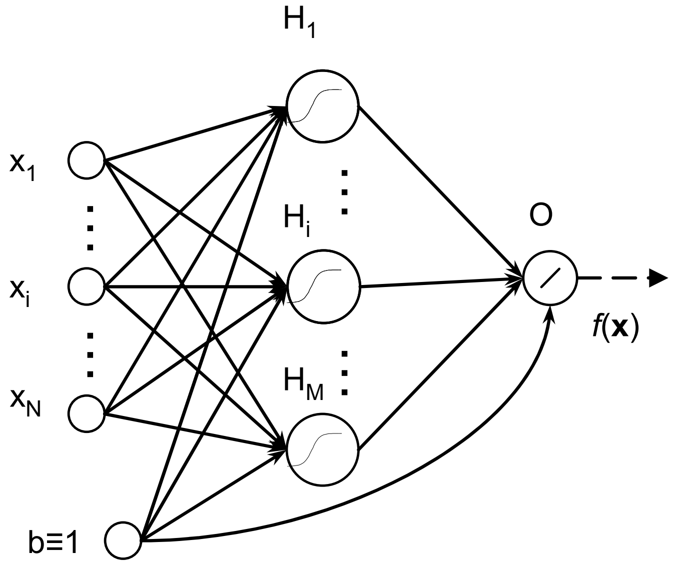


FIGURE 1.3 *A conventional multilayer neural network (MNN)*

During the ANN training, the network weights are modified to minimize a measure of the output error. If the considered output error measure is the mean square error (MSE) averaged over all the output units and the training set patterns, and the ANN is a FNN whose units have continuous and differentiable activation functions, the efficient and recursive error backpropagation (EB) training algorithm is applicable (Widrow & Lehr, 1990; Rumelhart et al., 1986).

Therefore, conventional MNNs have been traditionally trained using the standard EB, which implements the steepest descent rule to minimize the MSE surface in the weight space by iterative scanning each pattern of the training set. The training is generally arrested after a prefixed number of complete scanning of the training set (*epoch*), or when the output MSE goes below a prefixed threshold. The available input-output pairs may be also divided between a training set and a *test set* against which the MSE is periodically measured: in this case, the training will generally stop as soon as the MSE on the test set stops to decrease.

Notwithstanding their impressive diffusion, the following age-old claimed EB drawbacks have raised some skepticisms on the application of MNNs:

- the slow learning rate (Cybenko, 1989; Hornik et al., 1989), and
- the lack of physical meaning of the representation of the modeled system built during the training (Rumelhart et al., 1986).

Despite its numerical efficiency due to recursion, EB is a first order steepest descent which suffers from slow convergence due to its potential to zigzagging about the actual direction to a local minimum. More sophisticated second order variants may generally improve convergence rates (e.g. *quasi-Newton* algorithm) (Broohmhead & Lowe, 1988), whereas some known factors and corrections may be applied to the weights update equation to modulate the learning factor and/or introduce a memory of the past weight updates via the so called *momentum* (Rumelhart et al., 1986; Widrow & Lehr, 1990).

Notwithstanding that, the EB learning rate may be so slow to only allow the application of MNNs to "off-line" static tasks where training is performed once and for all, whereas the application to real-time adaptive systems, where repeated training may be required with strict time constraints, may be precluded (Moody & Darken, 1988).

Regarding the lack of physical meaning of the MNNs, during the learning an internal representation of the desired mapping regression task is built by the training of the network weights. The difficulty related to the physical meaning of this representation is due to its distributed nature, as it is the whole pattern of activity over all the hidden units, and not the meaning of any particular hidden unit, that is relevant. This is directly related to the non-limitation of the support of the sigmoidal activation functions (Rumelhart et al., 1986).

Among other secondary but not negligible drawbacks of MNNs, we also highlight:

- as each MNN unit needs the incoming outputs from all the preceding units to evaluate its activation, a total parallelism is not realizable in MNN implementations,
- the so called *negative interference* effect (Schaal & Atkeson, 1998).

As any weight update during learning only greedily reduces the error related to the current training data, MNNs usually show excellent interpolation power but poor generalization capability outside of the range of training data. Due to the non-local nature of sigmoidal basis function, any change to the MNN weights has non-local effect that may lead to the effect known as negative interference when the input-output relationship is not stationary. If a new training phase is required, the MNN updates its weights to fit the new data, but it may catastrophically lose accuracy in the previous range of data.

As in this work we propose to use ANNs to perform eye tracking, we investigate in the following new neural architectures and models to overcome the mentioned drawbacks of conventional MNNs.

In particular, we highlight that, as the other described drawbacks, also the slow learning rate of MNNs is due to the infinite support of the sigmoidal activations.

We thus focus the attention and the investigation on ANNs with compactly supported kernels that are thus potentially able to overcome all the mentioned drawbacks of MNNs.

Chapter 2

Feasibility of a new geometry-free eye tracking

ABSTRACT

Existing eye gaze tracking systems strongly depend on the system setup geometry included in the explicit model of the mapping from the eye features to the point of gaze, so that rigid positioning constraints for the user and the system components should be respected.

The most used pupil centre corneal reflection technique requires very complex illuminating and image capturing systems that generally need to be synchronized each other.

A new simple illuminating system generating only a triangular pattern of three glints avoids the synchronization with the image capturing system, whereas the use of artificial neural networks to directly evaluate the mapping function allows to not assume any explicit model, so giving a geometry-free system.

Following these rationales, a prototype of an eye gaze tracking system is built and successfully tested during several sessions of real operation, so proving the feasibility of the proposed approach[*].

[*] Results described in this Chapter were published in Proceedings of the National Congress of Bioengineering, pp. 657-658, 2010.

2.1. Introduction

The pupil-centre corneal reflection (PCCR) is the well known and commonly used non-intrusive technique for EGTSs: the POG is evaluated from the output of a video camera catching the user's eye illuminated by infrared light. The aim, in this Chapter, is to prove the feasibility of a new EGTS based on PCCR.

Existing EGTSs strongly depend on the system setup geometry included in the explicit model of the mapping from the eye features to the point of gaze. Therefore, very strict and rigid constraints for the positioning of user and the system components should be respected.

The PCCR technique requires very complex illuminating and image capturing systems that generally need to be synchronized each other.

A new simple illuminating system generating only a triangular pattern of three glints avoids the synchronization with the image capturing system.

Furthermore, the integration of that illuminating system with a mapping function based on artificial neural networks (ANN) allows the system to be geometry free, not assuming any model either for the eye geometry, or positioning constraints for the user and the system components, as other solutions do (Yoo & Chung, 2005; Zhu & Ji, 2007).

The Root Mean Square Error (RMSE) of the POG position has been evaluated on real data showing an accuracy of the same order of magnitude of other commercial EGTS, thus proving the feasibility of the proposed system.

2.2. Materials and Methods

According to the PCCR technique, the user's face is illuminated with harmless near Infrared Light Emitting Diodes (ILED) and the POG is evaluated from two important image features captured by a video-camera catching one of the user's eyes: – the *glint*, a fraction of the IR light reflected off the corneal surface, appearing in the camera as a small intense area, – and the *bright eye*, the reflection of the retina appearing in the camera as an area larger and less intense than the glint, but easily detected from the dark infrared image of the surrounding iris (similar to the reflection of visible light giving the red eye effect in photography). As long as the user's head remains stationary relative to the camera, the glint position remains fixed in the image field, whilst the bright eye follows the eye motion. The POG can be determined from the relative position of the bright eye and glint (Hutchinson et al., 1989).

This proposal is inspired by the PCCR technique: 3 sources of IR light, each formed by 3 ILED (Figure 2.1), project a triangular pattern of 3 glints on one of the user's eyes (Figure 2.2).

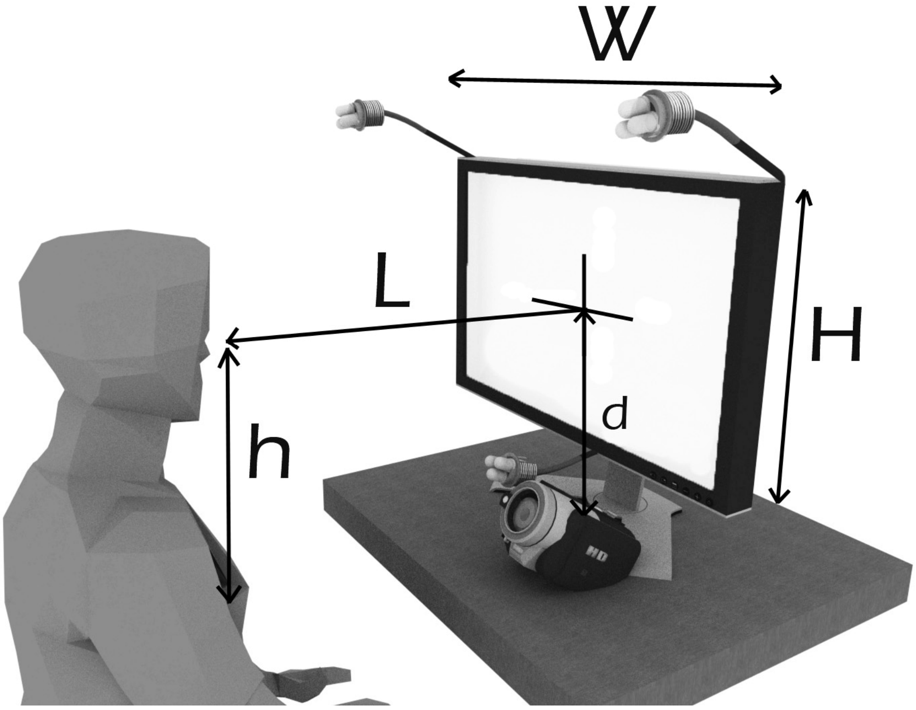


FIGURE 2.1 *The system typical setting*

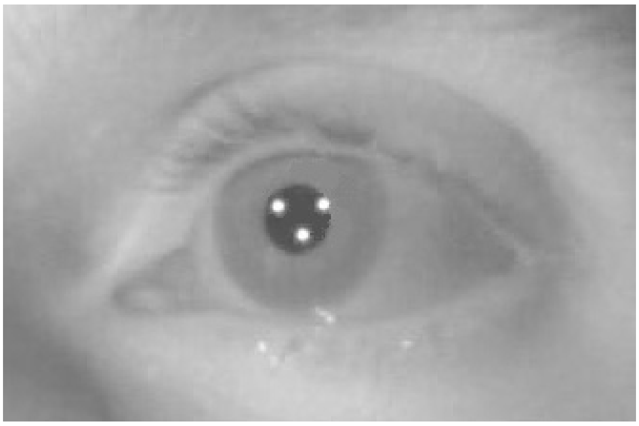


FIGURE 2.2 *The triangular pattern reflected by the eye*

As the 3 glints are the brighter points in the images, they can be easily detected by suitable thresholding. The pupil is also detected as a circular shape by means of the Hough transform (HT) as traditionally introduced in computer vision (Duda & Hart, 1972), with no need to provoke the above described bright eye effect.

The use of ANNs to map the eye features coordinates into the POG is not new (Baluja & Pomerleau, 1994; Piratla & Jayasumana, 2002). In this proposal the coordinates of the 3 glints and the pupil centre feed two multilayer ANNs, one for each of the POG coordinates (Figure 2.3). Regardless of particular positioning of the user and the system components, the ANNs can be trained to evaluate the POG allowing the system to be geometry free.

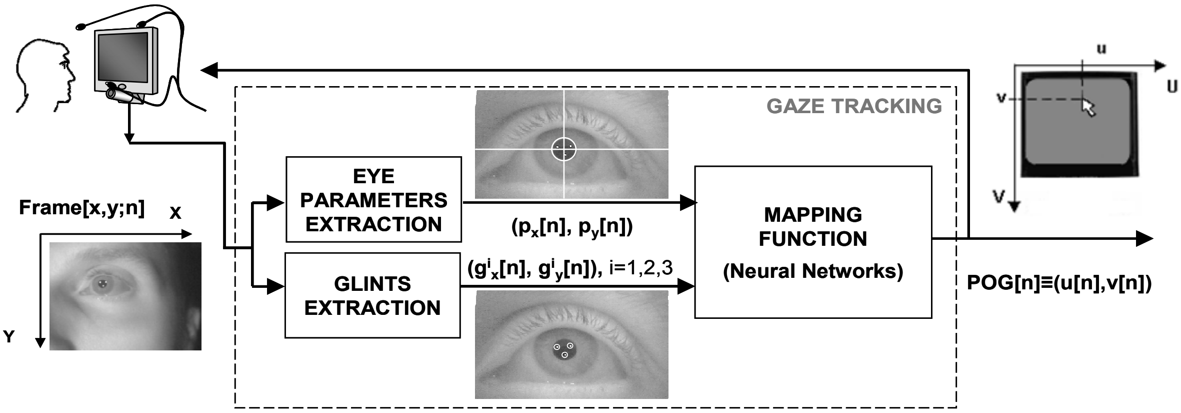


FIGURE 2.3 *The processing chain of the proposed system*

The used CMOS camera is adapted to capture IR light by replacing the original IR-Block filter with a black photographic negative film having IR-Pass and Visible-Block behavior. The camera is also provided of a 22x Zoom lens.

With reference to Figure 2.1, the test is conducted with the user positioned in front of a 23” 2048x1152 resolution LCD monitor (W=52 cm, H=28 cm, L=60 cm, h=36cm), whereas the camera is placed under the monitor (d=28cm), and the 3 light sources are freely placed pointing the user’s eye and forming approximately an equilateral triangle. The camera captured 30 fps with 640x480 spatial resolution.

The neural mapping function is trained with data acquired during an initial calibration phase, when the user stares at a sequence of known points in a 5x5 grid on a conventional monitor. The performance of the system is evaluated in terms of the RMSE of the positions – for the two different POG coordinates – when the user stares at 9 points located at intermediate positions respect to the calibration training grid. Two sessions, one for each of two different users, were performed.

2.3. *Results and Discussion*

As the scope of this preliminary work was only to prove the feasibility of the proposed system, a research neither on the best positioning of user and system components, nor on the ANN architecture and parameters optimization was performed during the test.

In the previously described test setting, the obtained RMSE is about 1.4° (Table 2.1). The results show an interesting substantial coincidence of the RMSE for the two components of the POG.

X Component RMSE		Y Component RMSE	
cm (inch)	**deg** (rad)	cm (inch)	**deg** (rad)
1.46 (0.58)	**1.40** (0.02)	1.46 (0.57)	**1.40** (0.02)

TABLE 2.1 *Root Mean Square Errors for the proposed eye gaze tracking system*

2.4. *Conclusions*

A preliminary prototype of EGTS inspired to PCCR technique was built following the main two guidelines given in Chapter 1 and regarding the use of a new IR illuminating system and a mapping function based on ANN.

The proposed EGTS is tested on users performing real sessions of operation and, even if neither the position of user and system components, nor the ANN architecture and parameters optimization was performed, an encouraging RMSE of about 1.4° was reached for both the coordinates of the POG.

These results not only proof the system feasibility, but also approaches the performance stated by commercial systems using more onerous hardware and superior data rate (i.e. LC Technologies Inc., Eyegaze Systems. [Online] reports an accuracy of 0.45° with L=51 cm, 60 fps with no expressly specified spatial resolution).

Chapter 3

Model independent and free geometry eye tracking

ABSTRACT

A new model-independent eye-gaze tracking system based on the pupil centre corneal reflection is proposed. The neural mapping function allows to avoid any specific model assumption and approximation either for the user's eye physiology or the system initial setup admitting a free geometry positioning for the user and the system components. The robustness of the proposed system is proven by assessing its accuracy when tested on real data coming from: i) different users; ii) different geometric settings of the camera and the light sources; iii) different protocols based on the observation of points on a calibration grid and halfway points of a test grid.

The achieved accuracy is not greater than 0.49°, 0.41°, and 0.62° for respectively the horizontal, vertical and radial error of the point of gaze. Then, the actual system performs better than eye-gaze tracking systems designed for human computer interaction which, even if equipped with superior hardware, show accuracy values in the range 0.6°-1°[*].*

[*] Results described in this Chapter have been submitted for Publication in *Journal of NeuroEngineering and Rehabilitation*.

3.1. Introduction

While a 1° accuracy is an agreed bound for the specifications of EGTSs designed as input devices for HCIs. Therefore, we test and tune the EGTS described in Chapter 2 aiming to reach the lower bound of 0.6° accuracy, coming from the physiological evidence that in the fovea, the highest acuity retinal area ranges from 0.6° to 1° (Guestrin & Eizenman, 2006).

The structure of the present Chapter is reported in the following: Section 3.2 describes the theoretical basics of the proposed EGTS, detailing information on each component in subsection 3.2.1, showing the setup and the experimental protocol in subsection 3.2.2. In particular, the robustness of the proposed EGTS is proven measuring its accuracy on real data captured from healthy subjects for different geometric settings of the system setup, considering not only the known points used during the calibration, but *also* halfway test points the user did not cross during the calibration. In subsection 3.2.3, details are provided regarding the performance evaluation metrics used: the proposed EGTS was compared in terms of performance with several model-independent (Baluja & Pomerleau, 1994; Piratla & Jayasumana, 2002; Zhu & Ji, 2004) and model-based (Guestrin & Eizenman, 2006; Villanueva & Cabeza, 2008) methods described in literature, which are briefly reviewed in subsection 3.2.4, together with two commercial EGTSs (LC Technologies Inc., Eyegaze Systems. [Online], Tobii Technology. [Online]).

In Section 3.3, results in terms of achieved accuracy are reported and discussed. The proposed EGTS performance met the requirement of 0.6° accuracy and was practically independent on both the system setup and the user. No noticeable training effect in using the system resulted.

As summarized in the concluding Section 3.4, the proposed EGTS generally bettered other above referenced model-independent and model-based methods in literature, approaching the performance of the mentioned commercial EGTSs equipped with superior hardware.

3.2. Materials and Methods

3.2.1. The proposed system basics and components

Reference (Guestrin & Eizenman, 2006) presented a general study for PCCR covering all the possible system configurations in terms of number and positioning of IR light sources and cameras. Although under general simplifications (corneal spherical approximation, light sources assumed as point sources, cameras assumed as pinhole cameras), some important results were found:

- *1 camera, 1 IR source*: the POG cannot be estimated unless the head is stationary or the head position is estimated by some other means,
- *1 camera, 2 IR sources*: is the simplest configuration that allows estimating the POG letting the head free.

Under similar assumptions, in (Villanueva & Cabeza, 2008) it is also showed that:

- regardless of how *many* cameras or IR sources are used, calibration is necessary,
- *1 camera, 2 IR sources*: is sufficient (about 1° of accuracy), whereas the use of more IR sources and calibration points increase the accuracy.

Considered the above results and the need to minimize the number of the inputs, we propose to use one camera and to increase the number of IR lights from two to three so that an opportune triangular pattern of glints is projected on the user's eye (Figure 3.1) allowing the POG estimation even when the head moves. It will be shown in the following that the triangular pattern of glints in Figure 3.1 allows convenient and robust eye feature detection.

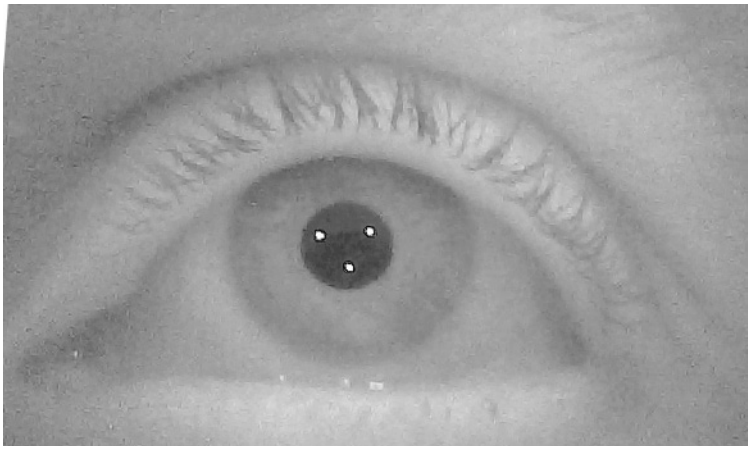


FIGURE 3.1 *The triangular pattern of the three glints reflected by the eye*

As depicted in Figure 3.2, the processing chain of the proposed EGTS starts with two separate blocks extracting the locations of the pupil centre and the three glints that feed two MNNs, one for each of the POG coordinates. The MNNs can be trained for whatever positioning of the user and the system components, allowing neglecting any system or subject-specific eye parameters measure/estimation (free geometry setup). The initial system setup is thus extremely simplified and the following measurements and procedures can be avoided:

- *camera calibration* (the determination of intrinsic camera parameters): any kind of camera can be used,
- *system geometry determination*: IR lights, user, monitor and camera can be freely positioned,
- *monitor measurement*: any kind of monitor can be used, regardless of the resolution and dimension,
- *user's eye physiology determination*: once the initial setup has been done, the system can be used by different users.

Moreover, whatever change should occur for the system configuration in terms of substitution or positioning of the components, no additional measurements or software modifications are needed. Any constraint to rigidly keep the system invariant after the initial setup may be thus relaxed.

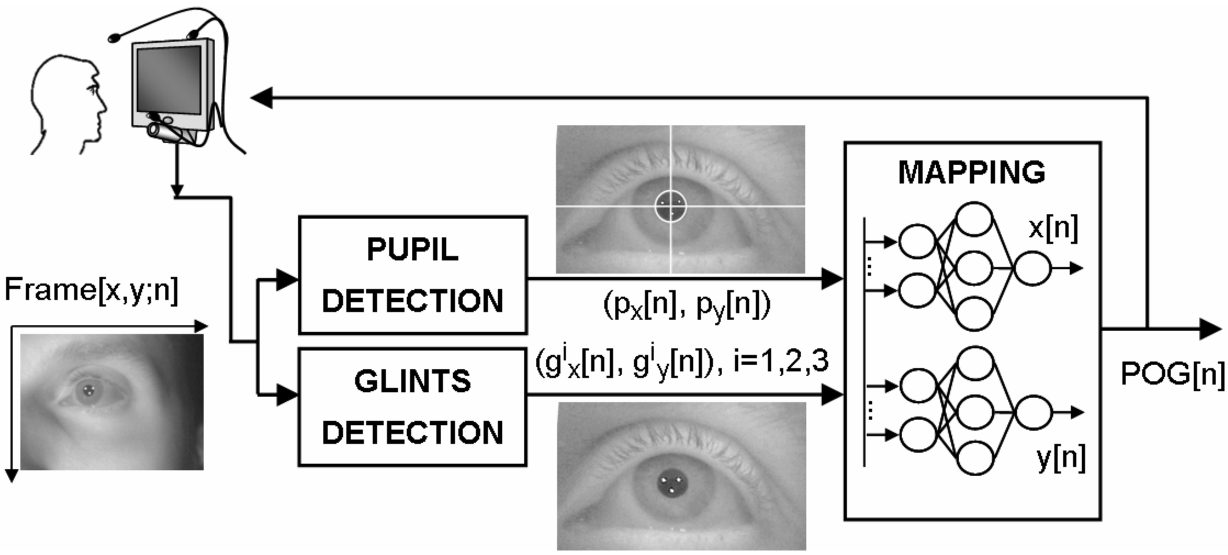


FIGURE 3.2 *The processing chain of the proposed system*

Experimental and simulation results in (Guestrin & Eizenman, 2006) suggested that even relatively small errors in the estimation of the pupil centre and glints can result in relatively large POG estimation errors. We thus provide in the following some details about the methods used to perform the features extraction.

Two pupil effects are mainly used to detect pupils: the so called *dark pupil* and the *bright-eye*, which have been briefly described in Section 1.2. Some solutions use both the mentioned effects requiring two or more on and off-axis light sources be multiplexed in time as well as in the wavelength and/or in the polarization (Ebisawa, 1998; Morimoto et al., 2000; Zhu & Ji, 2004). Time multiplexing requires synchronization between the camera frame rate and the light sources activation cycles and causes the POG estimate to be provided at half-rate of the camera frame rate.

In addition to the circuital complication due to the time multiplexing, other important limitations of the bright-eye effect are: its large variability among subjects; the evidence that from 5 to 10% of people have not sufficiently intense bright-eyes to allow reliable POG estimation (Hutchinson et al., 1989); the need to place light sources near the camera axis; and an uncontrolled variability of its effect led by even minor head rotations (Zhu & Ji, 2004; Hansen & Ji, 2010).

As other authors did (Droege et al., 2007), we opted to use only off-axis lighting and dark pupil to estimate the pupil centre, so to avoid the limitations of the bright-eye, reduce the circuital complexity, let free the positioning of the IR light sources and camera, let the head free to move, and ease future work including the use of two eyes (the two bright effects will be hardly the same).

As the proposed three ILEDs approach provides a sufficiently large contrast eye image, (the pupil is darker than its surroundings), a simple binary threshold can be successfully applied for the pupil detection. As we considered an indoor environment, the threshold value has to be initially set for each session and does not need to be adjusted during the same session. After the image thresholding, since we are interested in the centre of the pupil and not in the real pupil shape, the Hough transform for circle finding (Duda & Hart, 1972) is satisfactorily used as other authors did (Droege et al., 2007). To decrease the computational burden associated with the Hough transform, the preliminary binary thresholding speeds up the calculation and improves the precision of the pupil centre detection. When no pupil centre comes from the preceding frame, the whole actual frame is processed to find out possible circles. Frames for which the previous pupil centre is available, are processed only in a rectangular region of interest, reducing the computational load.

Reference (Villanueva & Cabeza, 2008) reported that the noise in glint position estimation is due to the glint reduced size, and this brought to the use of two of them. Moreover, the glint detection can be detrimentally affected by artifacts due to the glint rolling off the cornea onto the irregular sclera during large eye rotation (Torricelli et al., 2008), daylight, spurious reflections, and non-spherical curvature at the edges of the cornea. We thus propose to use a three glints pattern, which not only improves the POG accuracy (Morimoto, 2000; Villanueva & Cabeza, 2008), but also adds information by projecting onto the user's eye a known pattern (Figure 3.1) that can be used to detect and discard glint artifacts.

The glint detection is solved by a three-stage algorithm: first, the three glints-associated blobs are detected using a binary threshold; second, the centre of mass of each blob is calculated with subpixel accuracy, giving the glint candidates; third, some geometric relationships and heuristics related to the triangular reflected pattern are applied to discover and exclude possible artifacts:

- the direction of the three lines joining the three couples of glints candidates must be 0°, 60° and 120° ± some tolerance, otherwise the frame is discarded,
- each side length of the triangle formed by the glint candidates must fall within a specific range of values, otherwise the frame is discarded.

Despite their simplicity, the verification of the above conditions has been shown very powerful in identifying and discarding spurious glint artifacts.

The subpixel accuracy provided by the coordinates of the three glints and pupil centre detection stages is then profitably used in the overall training and neural mapping function.

The pupil centre and glints are indeed used to feed the ANNs in such a way to minimize the number of input neurons. Moreover, in order to minimize the number of output neurons, we propose the use of two separate MNNs, each one having the same eye features as inputs, with one single output neuron directly estimating one of the X and Y coordinates of the POG. The POG discrete coordinates given by the pixels of the screen will be thus given by the quantization of each of the two MNNs output. Regarding the training of the two MNNs, we propose a rectangular, uniform calibration grid to build an opportune training set, as previously described.

One hidden layer and the standard EB training algorithm is used, whereas the transfer function for the hidden layer and the output units are the hyperbolic tangent (tanh) and the linear function, respectively. The best parsimonious architecture using 10 hidden neurons was heuristically found as the best performing for both the MNNs.

3.2.2. The experimental setup and protocol

Some EGTSs aim at using low-quality (web) cameras to minimize costs. Low cost solutions with a standard lens may require the camera to be too close to the eye. We thus opted for an analogue B/W video-surveillance camera (FC II Computar, CS mount, Senview varifocal 6-60mm lens with AutoIris). The camera is connected to a frame grabber

(EASYCAP DC60, 25 fps) through its composite video output. The OpenCV software framework used to perform the image processing phase samples each frame giving 480×640 pixels. In front of the camera, a Perspex IR-pass/visible-block filter (wavelengths under 780nm are blocked) was placed. The overall cost of the described optical system was under 200 € so giving a low cost solution. The triangular off-axis illuminating system was obtained using a three-arm flexible support built with simple twisted wire supporting three groups of four USB-powered ILEDs.

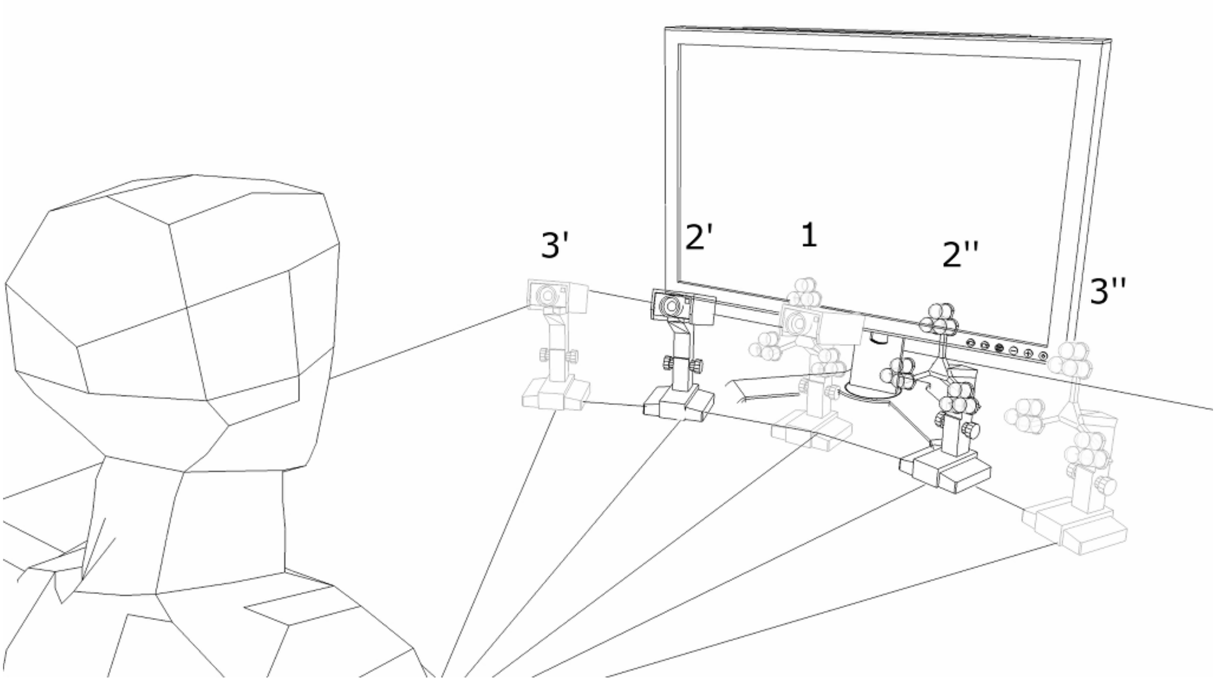


FIGURE 3.3 *The three different geometric settings for the system setup*

The tests were conducted by positioning the user in front of a 17" monitor (1024×768 spatial resolution and 4:3 aspect ratio) 70 cm far from the user's eye. In order to assess the independence of the proposed EGTS from the geometry, the accuracy of the POG estimation was evaluated for three different geometric settings depicted in Figure 3.3. The camera was never calibrated and always placed under the monitor.

In the first setting, the triangular lighting system was placed around the camera (see 1 in Figure 3.3). In the second setting, the camera was placed at an angular distance of 15° to the left of the monitor (see 2' in Figure 3.3) and the lighting system was placed 15° to the right (see 2'' in Figure 3.3) so that the overall angular displacement between the camera and IR lights centre was 30°. The third setting was similar to the second one but the overall angular displacement between the camera and IR lights was 60° (see 3' and 3'' in Figure 3.3).

A 4×5 calibration grid of uniformly spaced points was chosen as it uniformly samples the 4:3 aspect ratio screen (Figure 3.4, left). The 3×4 test grid is given by the halfway points of the calibration grid (Figure 3.4, right): it is here outlined that the user's gaze never crosses the points on the test grid during the calibration. Accuracy was evaluated during both the calibration and the test phases.

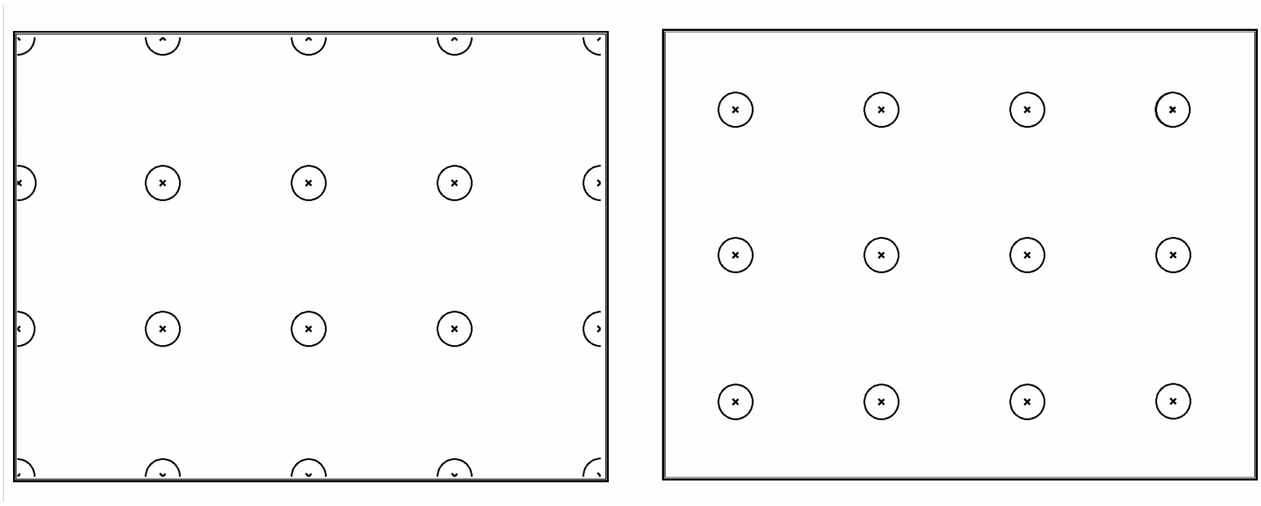


FIGURE 3.4 *The 4×5 calibration grid (left) and the 3×4 test grid (right)*

Two consecutive test sessions were performed for each of six healthy volunteers, participating to all of the three mentioned geometric setting sessions, proposed in random order. Each session directly started asking the user to fix her/his gaze to each calibration point for a fixed period of 1200 ms, corresponding to 30 frames at a capturing rate of 25 fps. During the calibration, the MNN training set was built collecting only the input given by the estimated centers of the glints and pupil without collecting any image frame. The corresponding desired outputs are given by the coordinates of the known calibration points. The MNNs training started after the calibration and lasted 1000 epochs. The user was then asked to fix her/his gaze upon each point of a pseudo-random sequence of points on the test grid. Each test point was showed five times, each time for a fixed period of 600 ms (corresponding to 15 frames). The protocol was described to each user letting her/him alone and unassisted during the fully automatic overall calibration and test procedure. Each user freely chose the used eye.

Even if the use of ANNs-based mapping functions was showed able to incorporate head movements into the mapping as in (Baluja & Pomerleau, 1994; Piratla & Jayasumana, 2002; Zhu & Ji, 2004), in this preliminary analysis we opted to defer to future work the tuning and tweaking of the calibration phase to evaluate the performance of the proposed EGTS when users are let free to naturally move their heads. The users were thus asked to keep the head still by means of a head/chin-rest. This also avoided the users to get out from field of view and/or out of focus of the camera

3.2.3. Performance measurements and evaluation criteria

Human vision and EGTS accuracy is usually expressed in terms of the angular error in visual degrees (smaller angle means higher accuracy) in order to be independent from the screen resolution and distance from the user. Given the angular accuracy, the error projection can be easily derived for each distance between the user and the POG surface using obvious trigonometry (Figure 3.5).

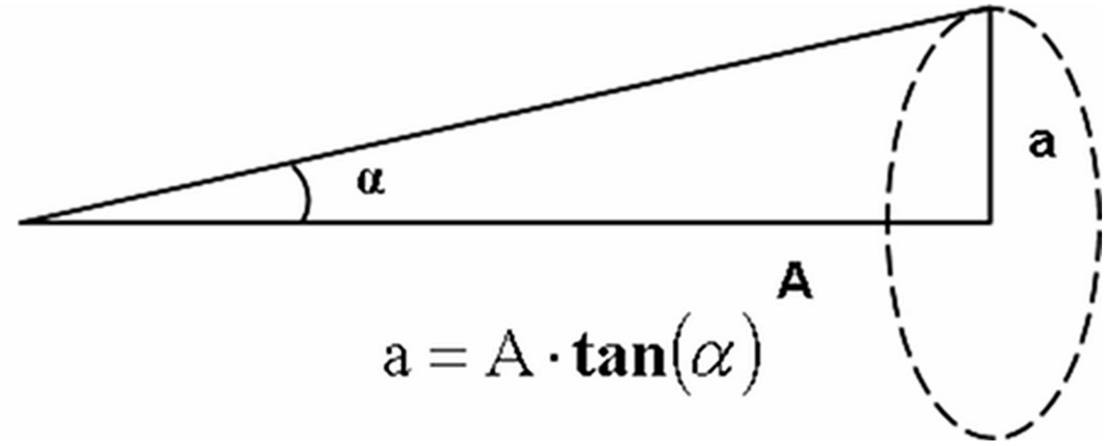


FIGURE 3.5 *Visual angle trigonometry*

Although the human eye is commonly considered a highly accurate sensor, if used as an input device the exact POG location is inherently not as precise as with a mouse (Duchowski, 2002).

As a matter of fact, when the user is gazing at a particular point, her/his eyes are oriented in such a way that the POG projects itself on the fovea (the highest acuity region of the retina). Even if during the visual fixation on a still object, the POG is perceived as fixed, it is not. This is done to prevent the complete fading of vision, giving blindness during visual fixation (Martinez-Conde et al., 2004).

Moreover, the fovea small retinal area is projected onto a finite visual angle (from 0.6° to 1°, (Guestrin & Eizenman, 2006)), and when we move the eye in order to place the fovea on the area that we want to see with fine details, we do not need to place it exactly centered and on top of the fovea as its projected area becomes larger and hence, covers more the further away an object is (Tobii Technology. [Online]).

Given the above considerations, it follows that a visual fixation can thus be defined as a stable position of the POG that presents visual angle dispersion below 1° (foveal area upper limit). POG estimation errors below 1° are thus pursued by most EGTS designers (Villanueva & Cabeza, 2008).

We may instead add that if the EGTS is designed for HCI it is worthwhile to achieve accuracy under the usually accepted 1° requirement, trying to approach the mentioned lower value of 0.6° for the fovea visual angle. The accuracy showed by the proposed EGTS will be thus analyzed in next Section 3.3 considering the lower limit values of 0.6°.

For the generic n^th frame during which the user is gazing at one of the known points of either the calibration or test phases, the quadratic error $e^2[n]$ between the known point position and the POG estimation was evaluated and accumulated for both the X and Y coordinates. The mean squared error (MSE) and the root mean square error (RMSE) in equations 5.1 were thus evaluated averaging the error along the N frames of each phase.

$$\begin{aligned} &\mathrm{MSE_x} = (\textstyle\sum_n e_x^2[n])/\mathrm{N}, \\ &\mathrm{MSE_y} = (\textstyle\sum_n e_y^2[n])/\mathrm{N} \qquad (5.1) \\ &\mathrm{RMSE_x} = (\mathrm{MSE_x})^{1/2}, \\ &\mathrm{RMSE_y} = (\mathrm{MSE_y})^{1/2} \end{aligned}$$

As observed in (Villanueva & Cabeza, 2008), the results given in terms of the RMSE for X and Y coordinates do not properly measure the POG estimation error. Rather, these errors highlight differences in horizontal and vertical coordinates. The Euclidean distance in equations 5.2 between the real and estimated POGs should be considered as the most representative error value.

$$\begin{aligned} &\mathrm{MSE_\rho} = (\textstyle\sum_n(e_x^2[n]+ e_y^2[n]))/\mathrm{N} = \mathrm{MSE_x} + \mathrm{MSE_y} \qquad (5.2) \\ &\mathrm{RMSE_\rho} = (\mathrm{MSE_\rho})^{1/2} \end{aligned}$$

The errors in equations 5.1 were evaluated in terms of pixel difference and then converted into degrees by using the visual angle trigonometry in Figure 3.5. The Euclidean $\mathrm{RMSE_\rho}$ was evaluated as in equations 5.2.

3.2.4. Other known systems

For the sake of comparison, in this section we briefly consider several relevant EGTSs against which the proposed EGTS was tested. Firstly, we report a few details about some model-independent EGTSs based on ANNs. Then we describe some model-based methods. Lastly, we shortly review two commercial systems currently giving the *de facto* accuracy lower bounds for EGTSs used as HCI. For simple reference and comparison, the important topics are summarized in Table 3.1.

Cameras	Lights	Calibration	Approach (mapping function)	Accuracy	Reference	Comments/Notes
1 20 fps, 640×480	1 (+1)	screen divided in 2×4 zones	MI (2 GRNNs)	5°H 8°V	(Zhu & Ji, 2004) [abc]	Two concentric IR light rings are alternately turned on and off. Only one glint is used.
1 30 fps, 640×480	1 (0)	12×16 grid	MI (1 MNN)	2.4°H 2.4°V	(Piratla & Jayasumana, 2002) [bc]	Special spectacles frame is needed; both the eyes are used. Accuracy is measured on testing points.
1 low resution	1	cursor moves	MI (2 MNNs)	1.5°	(Baluja & Pomerleau, 1994) [abc]	Accuracy is measured on testing points.
1 30 fps, 640×480	2	3×3 grid	MB	0.9°	(Guestrin & Eizenman, 2006) [ac]	
2	2		MB	0.68°	(Guestrin & Eizenman, 2006) [ac]	Preliminary simulations
1 60 fps, 640×480	2-4	1 point	MB	≈ 0.7°H ≈ 0.7°V ≈ 1°	(Villanueva & Cabeza, 2008) [ac]	Each calibration point produces a gaze estimation model. 17 one-point calibrations were performed.
1 60/120 fps, 640×480	4 (+1)	5 points	MB	0.5°	(Tobii Technology) [ad]	Tracks both eyes simultaneously. Camera and the IR sources are built in the monitor.
1 60/120 fps	1 (+1)	9 points	MB	0.45°-0.70°	(LC Technologies) [ad]	Typical and worst accuracy is reported. The POG estimation may require a dedicated computer.
1 25 fps, 640×480	3	4×5 grid	MI	≤ 0.41° H ≤ 0.49° V ≤ 0.62° E	(Proposed EGTS) [abc]	Worst accuracy bounds measured on halfway 3×4 testing grid for 3 different geometric settings.

The "cameras" column shows the number of cameras used by the method and the cameras characteristics (frames per second and spatial resolution expressed in terms of the width × height number of pixels). The "lights" column shows the number of light sources necessary for the methods. An additional "+1" means that an extra alternately switched on-axis light sources is used to generate bright eye and not to generate glints. An additional "0" means the light source is optional. The "approach" column corresponds to the approach adopted by the EGTS: model-based (MB) or model-independent (MI) method category. The "accuracy" column indicates the achieved POG estimation error. If available, the angular error is reported and expressed in degrees. When applicable the component of the error is indicated as horizontal component (H), vertical component (V), Euclidean distance (E). [a]Based on Pupil Corneal Reflection. [b]Mapping function based on artificial neural networks. [c]System proposed in scientific literature. [d]Commercial system.

TABLE 3.1 *Time series forecasting: generally used training set*

In (Baluja & Pomerleau, 1994), low resolution images were used and all of the 600 (15×40) pixels of a rectangular window surrounding the user's eye are used as input of two

MNNs. The outputs of each MNN are respectively provided by 50 units for the X coordinate, and other 50 units for the Y coordinate (the highest output unit represents the estimated coordinate). During the calibration, the user visually tracks a cursor moved in a pre-defined zigzag horizontal path on the screen, and each of the images of the eye is paired with the coordinates of the cursor giving 2000 image/position pairs gathered for training. Other 2000 image/position pairs were also gathered for testing. The best angular accuracy the system achieved on the 2000 testing points was 1.5°.

In (Piratla & Jayasumana, 2002), the user is requested to wear a particular spectacles frame to provide a reference fixed with the head. The EGTS is not PCCR-based and the used lamp is not essential. The coordinates of two points on the spectacle frame, two eyeballs centers and upper and lower eyelids provide the 12 inputs of the used MNN, whereas the X and Y POG coordinates are its 2 outputs. A 12×16 calibration grid is used and the estimated POG falls almost accurately in a 2×2 square inches window on the screen at distance between 30 and 60 cm (the visual angle trigonometry in Figure 3.5 gives a best accuracy of about 2.4° in both the directions).

In (Zhu & Ji, 2004), two identical generalized regression neural networks (GRNNs) – each with a single output unit – estimate the X and Y POG coordinates respectively. The two components of the pupil-glint vector, two coordinates of the single glint, the ratio of the major to minor axes and the orientation of the pupil ellipse provide the 6 inputs of the two GRNNs. During the calibration the user's gaze was quantized into 8 regions on the screen (2×4 grid) and the same gaze classification was performed by the two GRNNs outputs. The method achieved accuracies around 5° and 8° in the horizontal and vertical direction, respectively.

Some model-dependent EGTSs are now briefly described.

Reference (Guestrin & Eizenman, 2006) presented a general theory for PCCR-based EGTSs covering whatever cameras and IR light sources number and positioning, under the approximations adopted by most part of the model-dependent EGTSs (IR lights assumed as point sources, video cameras assumed as pinhole cameras, and cornea assumed as a spherical mirror). Test results were reported using a 9 point 3×3 uniform calibration grid for two system configurations, the first using one camera and two lighting sources, the second using an additional camera. Accuracies of 0.9° and 0.68° was respectively achieved.

Under similar assumptions, in (Villanueva & Cabeza, 2008) a geometric model based on glint positions and pupil ellipse was used to show the minimal required number of cameras, light sources, and user calibration points (user calibration was also showed unavoidable).

We now report some details about the two commercial EGTSs presented in (LC Technologies Inc., Eyegaze Systems. [Online], Tobii Technology. [Online]). Both the systems adopt a PCCR model-based method, use ILEDs, remote cameras, implement in software the overall processing, and require a user calibration to learn the radius of curvature of the cornea and the angular offset between the visual and optical axes of the user.

The EGTS in (Tobii Technology. [Online]) uses a built-in 640×480 resolution camera capturing two images of the eyes simultaneously at 60 fps or 120 fps producing the respective pupil and glints so providing the EGTS with two different sources of information. Three off-axis light sources are built in the monitor upper frame, whereas a fourth off-axis light source and an extra on-axis light source given by 2 concentric rings of ILEDs are placed around the camera. The EGTS requires a 5 points calibration during which both the bright and dark pupil effects are tested and the best method is chosen. The typical achieved accuracy is 0.5°.

In (LC Technologies Inc., Eyegaze Systems. [Online]) a 60 or 120 fps camera (no retrievable resolution) is located below the monitor and an ILED at the center of the camera lens generates the glint and the bright pupil. Reported typical and maximum average accuracy is 0.45° and 0.70°, respectively.

The hardware equipment related to both the two commercial EGTSs (LC Technologies Inc., Eyegaze Systems. [Online], Tobii Technology. [Online]) appears quite sophisticated and seems to be one of the reasons of their relatively high cost.

For the sake of completeness, the last row of Table 3.1 anticipates the performance achieved by the proposed EGTS that will be analyzed in the following section.

3.3. Results and Discussion

The RMSE for the Euclidean, horizontal and vertical coordinates for the three considered system settings are respectively reported in Table 3.2, Table 3.3, and Table 3.4.

The analysis of the results may start from the mean Euclidean RMSE_ρ: the overall mean RMSE_ρ averaged along the users and the sessions for the three system settings (third-to-last rows of Table 3.2, Table 3.3, and Table 3.4) is not only better than the generally accepted accuracy requirement of 1°, but it is also practically always under the limit of the 0.6° lower bound given by the human fovea, as previously discussed. The only exception is the 0.622°

RMSE_ρ (third-to-last row, last column of Table 3.3) related to the test grid of the first system setting, that is just slightly above the 0.6° threshold.

That proofs the validity of the proposed model-independent approach, in particular if we compare the performance of the proposed EGTS with the accuracy of systems summarized in Table 3.1. Only the two commercial EGTSs (LC Technologies Inc., Eyegaze Systems. [Online], Tobii Technology. [Online]) using superior hardware and rigidly assembled equipment declare a typical accuracy slightly better than the proposed EGTS.

Among the model-based EGTSs, the second system proposed in (Guestrin & Eizenman, 2006) achieved an accuracy of 0.68°, slightly worse than the proposed EGTS, but two cameras are required. All the other EGTSs reported in Table 3.1 performed worse than the proposed EGTS.

The substantial equivalent accuracy showed for all the three system settings also proofs the robustness of the proposed EGTS with respect to the geometry of the system setup.

A quite small inter-user Relative Standard Deviation (RSD) of the RMSE_ρ is showed for all the three system settings, ranging from the 8.5% of the test grid of the first setting (last row, last column of Table 3.2), to the 15.7% of the calibration grid of the third setting (last row, fourth column of Table 3.4). This demonstrates the robustness of the proposed EGTS with respect to different users.

The analysis of the results may follow with the examination of the error statistics related to each session: subjects no. 2 and no. 3 were practiced with the proposed EGTS, while the remaining subjects had no experience with it. Subjects no. 3 and no. 4 were shortsighted and even if the used eyes required almost 2 diopters of correction, no spectacles were worn during the tests. The performance of each user showed substantial coherence both for the three geometric settings and for the two consecutive sessions (e.g. users no. 1 and no. 2 were generally the best performers, whereas users no. 5 and no. 6 were generally the worst performers).

The overall mean accuracy and the accuracy achieved by single users for the two consecutive sessions are consistent, thus proving the absence of a noticeable learning effect, so that no particular training is required to effectively use the proposed EGTS.

Although the accuracy evaluated on the calibration grid is often slightly better than the accuracy on the test grid, their values may be practically considered equivalent.

That proofs that the proposed EGTS performs uniformly all over the screen and that the training of the ANNs giving the mapping is optimal.

The former point is also well shown in Figure 3.6 and Figure 3.7, respectively depicting the POG estimation clouds around each correct point on both the calibration and the test grid for the first session of the user no. 2. (please remember each point on the test grid is randomly shown 5 times, whereas each point on the calibration grid is shown just once). This interesting property grants that the proposed EGTS performs uniformly over the whole screen and does not suffer the quick fall off of the accuracy when the POG moves away from the calibration points as other EGTS generally do.

The point regarding the optimal neural learning grants that the ANNs realized the best approximation of the ideal mapping function. We used one hidden layer MNN and a trial and error approach selects the parsimonious architecture using 10 hidden units. This architecture showed the results reported in Table 3.2, Table 3.3, and Table 3.4, and required a training time compatible with a real-time operation. Other non reported results regarding using bigger ANNs models (up to 100 hidden units) gave unacceptable training time with no substantial gain in term of accuracy.

This latter point showed that the performance of the proposed EGTS is predominantly driven by the noise affecting the estimations of the PCCR eye features, confirming the conclusion other authors found performing error analysis (Guestrin & Eizenman, 2006).

Better accuracy can thus be obtained by means of superior hardware, and/or more sophisticated procedures to extract eye features.

Good correspondence was showed by the mean RMSE values achieved for the horizontal and vertical errors.

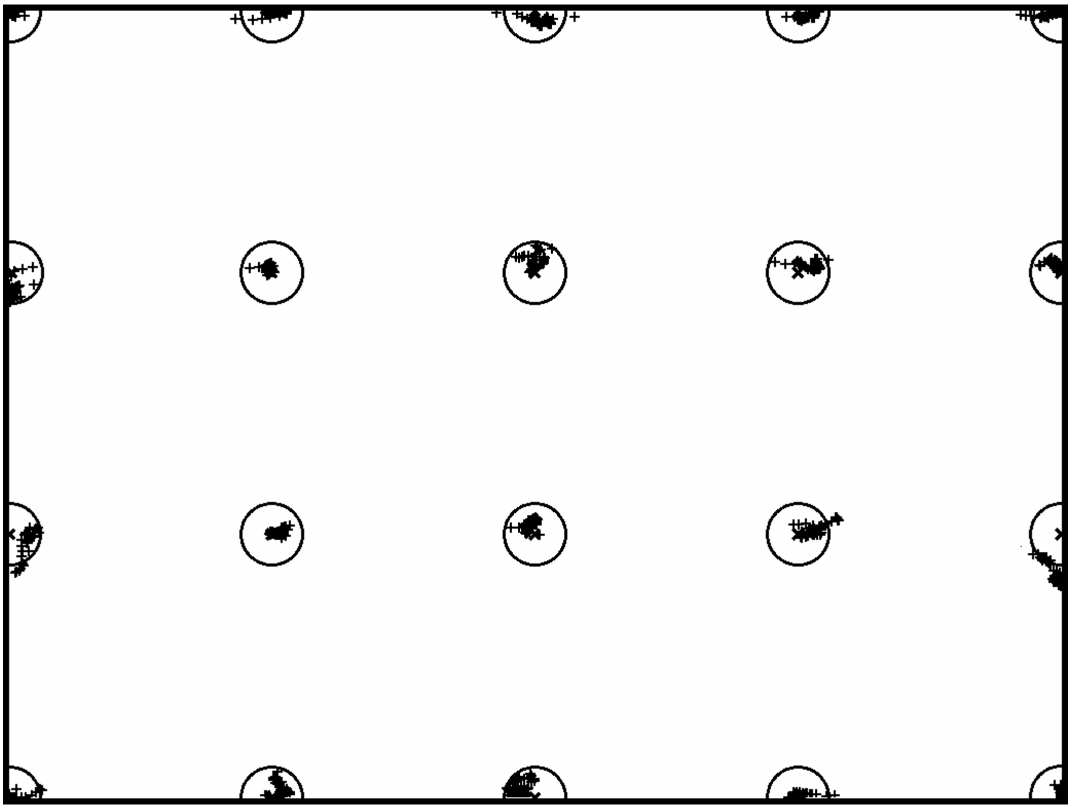


FIGURE 3.6 *Error distribution on the calibration grid (subject no. 2, 1st session)*

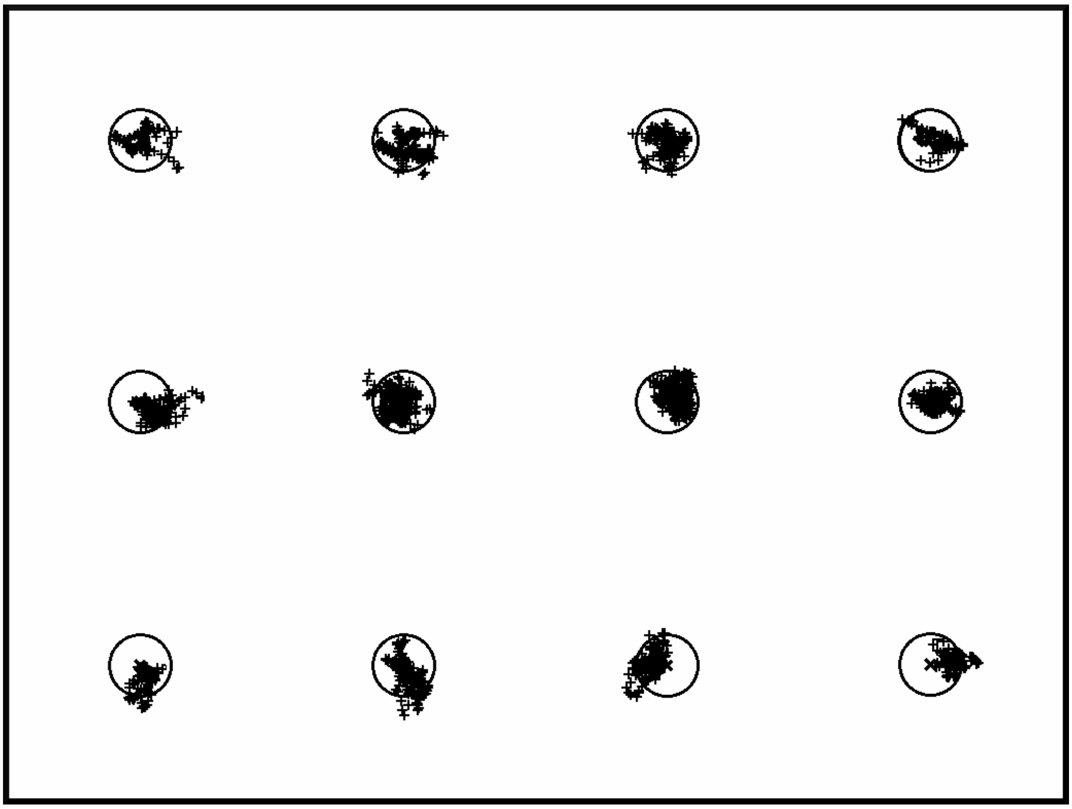


FIGURE 3.7 *Error distribution on the test grid (subject no. 2, 1st session)*

SYSTEM ACCURACY - **0°** BETWEEN IR LIGHTS AND CAMERA

Session 1	Calibration grid			Test grid		
User	RMSE$_x$	RMSE$_y$	RMSE$_\rho$	RMSE$_x$	RMSE$_y$	RMSE$_\rho$
1	0.378°	0.303°	0.485°	0.298°	0.526°	0.605°
2	0.336°	0.299°	0.449°	0.308°	0.414°	0.516°
3	0.361°	0.458°	0.583°	0.359°	0.498°	0.614°
4	0.415°	0.434°	0.601°	0.284°	0.612°	0.675°
5	0.414°	0.392°	0.570°	0.330°	0.466°	0.571°
6	0.444°	0.428°	0.617°	0.487°	0.480°	0.684°
mean	0.391°	0.386°	**0.551°**	0.344°	0.499°	**0.611°**
SD	±0.037°	±0.063°	±0.062°	±0.068°	±0.061°	±0.058°
RSD%	±9.4%	±16.3%	**±11.2%**	±19.8%	±12.2%	**±9.5%**
Session 2	Calibration grid			Test grid		
User	RMSE$_x$	RMSE$_y$	RMSE$_\rho$	RMSE$_x$	RMSE$_y$	RMSE$_\rho$
1	0.305°	0.512°	0.596°	0.443°	0.516°	0.680°
2	0.387°	0.460°	0.601°	0.421°	0.420°	0.595°
3	0.365°	0.494°	0.615°	0.336°	0.462°	0.571°
4	0.409°	0.351°	0.539°	0.426°	0.436°	0.610°
5	0.425°	0.429°	0.604°	0.482°	0.486°	0.685°
6	0.372°	0.470°	0.600°	0.389°	0.536°	0.662°
mean	0.377°	0.453°	**0.592°**	0.416°	0.476°	**0.634°**
SD	±0.038°	±0.053°	±0.025°	±0.046°	±0.041°	±0.044°
RSD%	±10.1%	±11.6%	**±4.2%**	±10.9%	±8.7%	**±6.9%**
mean	0.384°	0.419°	**0.572°**	0.380°	0.488°	**0.622°**
SD	±0.038°	±0.067°	±0.051°	±0.068°	±0.053°	±0.053°
RSD%	±9.9%	±16.0%	**±9.0%**	±18.0%	±10.9%	**±8.5%**

TABLE 3.2 *System accuracy of the proposed eye-gaze tracking system, 1st configuration: 0° between IR lights and camera*

SYSTEM ACCURACY - **30°** BETWEEN IR LIGHTS AND CAMERA

Session 1	Calibration grid			Test grid		
User	RMSE$_x$	RMSE$_y$	RMSE$_\rho$	RMSE$_x$	RMSE$_y$	RMSE$_\rho$
1	0.308°	0.322°	0.446°	0.333°	0.407°	0.526°
2	0.262°	0.329°	0.421°	0.355°	0.316°	0.475°
3	0.366°	0.424°	0.561°	0.359°	0.480°	0.600°
4	0.317°	0.432°	0.536°	0.355°	0.454°	0.576°
5	0.469°	0.424°	0.632°	0.409°	0.553°	0.687°
6	0.449°	0.395°	0.598°	0.521°	0.468°	0.701°
mean	0.362°	0.388°	**0.532°**	0.389°	0.446°	**0.594°**
SD	±0.075°	±0.046°	±0.076°	±0.064°	±0.073°	±0.081°
RSD%	±20.7%	±11.8%	**±14.4%**	±16.3%	±16.3%	**±13.6%**
Session 2	Calibration grid			Test grid		
User	RMSE$_x$	RMSE$_y$	RMSE$_\rho$	RMSE$_x$	RMSE$_y$	RMSE$_\rho$
1	0.249°	0.383°	0.457°	0.381°	0.362°	0.526°
2	0.328°	0.357°	0.485°	0.419°	0.369°	0.558°
3	0.383°	0.450°	0.591°	0.361°	0.436°	0.566°
4	0.306°	0.313°	0.438°	0.526°	0.361°	0.638°
5	0.383°	0.450°	0.591°	0.361°	0.436°	0.566°
6	0.462°	0.371°	0.592°	0.517°	0.419°	0.666°
mean	0.352°	0.387°	**0.526°**	0.427°	0.397°	**0.586°**
SD	±0.067°	±0.049°	±0.067°	±0.069°	±0.034°	±0.049°
RSD%	±19.2%	±12.7%	**±12.8%**	±16.2%	±8.4%	**±8.3%**
mean	0.357°	0.388°	**0.529°**	0.408°	0.422°	**0.590°**
SD	±0.071°	±0.048°	±0.072°	±0.069°	±0.062°	±0.067°
RSD%	±20.0%	±12.3%	**±13.6%**	±17.0%	±14.6%	**±11.3%**

TABLE 3.3 *System accuracy of the proposed eye-gaze tracking system, 2nd configuration: 30° between IR lights and camera*

SYSTEM ACCURACY - **60°** BETWEEN IR LIGHTS AND CAMERA

Session 1	Calibration grid			Test grid		
User	RMSE_x	RMSE_y	RMSE_ρ	RMSE_x	RMSE_y	RMSE_ρ
1	0.305°	0.328°	0.448°	0.433°	0.259°	0.505°
2	0.426°	0.382°	0.573°	0.513°	0.495°	0.713°
3	0.279°	0.367°	0.461°	0.347°	0.350°	0.493°
4	0.386°	0.479°	0.615°	0.281°	0.511°	0.583°
5	0.546°	0.352°	0.649°	0.487°	0.429°	0.649°
6	0.537°	0.440°	0.694°	0.488°	0.506°	0.703°
mean	0.413°	0.391°	**0.573°**	0.425°	0.425°	**0.608°**
SD	±0.103°	±0.052°	±0.092°	±0.084°	±0.093°	±0.088°
RSD%	±24.9%	±13.3%	**±16.0%**	±19.8%	±22.0%	**±14.4%**
Session 2	Calibration grid			Test grid		
User	RMSE_x	RMSE_y	RMSE_ρ	RMSE_x	RMSE_y	RMSE_ρ
1	0.336°	0.307°	0.455°	0.434°	0.342°	0.552°
2	0.348°	0.291°	0.453°	0.304°	0.388°	0.493°
3	0.298°	0.406°	0.504°	0.289°	0.475°	0.556°
4	0.367°	0.286°	0.465°	0.374°	0.397°	0.546°
5	0.440°	0.358°	0.567°	0.476°	0.474°	0.672°
6	0.433°	0.452°	0.626°	0.524°	0.490°	0.717°
mean	0.370°	0.350°	**0.512°**	0.400°	0.427°	**0.589°**
SD	±0.051°	±0.062°	±0.065°	±0.086°	±0.055°	±0.078°
RSD%	±13.8%	±17.7%	**±12.6%**	±21.5%	±12.9%	**±13.3%**
mean	0.392°	0.371°	**0.542°**	0.413°	0.426°	**0.598°**
SD	±0.084°	±0.061°	±0.085°	±0.086°	±0.077°	±0.084°
RSD%	±21.5%	±16.4%	**±15.7%**	±20.8%	±18.0%	**±14.0%**

TABLE 3.4 *System accuracy of the proposed eye-gaze tracking system, 3rd configuration: 60° between IR lights and camera*

3.4. Conclusions

Model-based approach to EGTS is analyzed and its drawbacks highlighted (oversimplified models, complex initial setup, and scarce flexibility of the system after the setup). A model-independent EGTS based on the optimal use of ANNs (training set and complexity of the architecture adequate to the POG estimation task) is proposed and realized. Large flexibility to different users, system setting, and a simplified free geometry setup is allowed, with no need to calibrate the camera and to perform any preliminary estimation or measure. That enables a relatively free engineering of the prototype giving large flexibility to both the assembly of the components and the potential applications.

The proposed EGTS showed also uniform accuracy all over the observed screen and neither particular training nor user assistance was showed needed.

The worst value of the achieved accuracy (0.622°) is quite better than the requirement of 1° usually accepted to design EGTSs to be used as HCI, approached the lower bound of 0.6° given by the projection of the human fovea, and proved the validity of the proposed model-independent approach.

Only commercial EGTSs using superior hardware and rigidly assembled equipment declare a typical accuracy slightly better than the proposed EGTS. The latter performs generally better than other examined model-based and model-independent systems.

As the use of ANNs was reported able to incorporate head movement into the EGTS mapping function, we plan to adequate the calibration phase by asking the user to opportunely move her/his head so to measure and achieve good accuracy even when the user is let free to naturally move it.

Future work is also planned to sophisticate the ANNs, for example using feedback connections, which should preserve some of the information from previously estimated eye features and POGs (e.g. recurrent networks).

The simplification and optimization of the calibration phase by minimizing the number of points, the gaze duration and the grid structure is another potential field of future investigation.

Chapter 4

Real-time adaptive neural predictors

ABSTRACT

The feasibility of eye gaze tracking with neural based mapping function encouraged the research on new neural networks architectures and learning schemes. In order to overcome the problems due to failures in eye features detection and head motion, a real-time time series prediction based on the neural networks used to regress the mapping function is proposed.

That prediction scheme is successfully validated applying it to the gesture recognition considering the time series given by the output of two accelerometers placed on the upper arm and on the forearm respectively. The prediction errors are used both to train the neural networks and to estimate a measure of the unlikelihood of the specific gesture occurrence. The first repetition of each gesture trains the related neural networks and the current motion is recognized after a few successive repetitions. Neither a priori assumptions nor signal pre-processing is performed. The training is performed at the beginning and can be repeated during the running (adaptability). On the four significant gestures considered (Wolf motor function test), the proposed method shows a correct recognition rate higher than 83%[*]*.*

[*] Results described in this Chapter were published in Proceedings of the 11th International Congress of the IUPESM - Medical Physics and Biomedical Engineering, Munich, Germany, vol. 25/IX, pp. 536-539, 2009.

4.1. Introduction

The aim, in this Chapter, is to prove the feasibility of a real time limb gesture recognition system based on ANNs.

The system is integrated with a simplified sensor set of only two accelerometers placed on the upper limb and the recognition rate is assessed on real data.

Due to their pattern classification and pattern recognition capabilities and their property to learn from and generalize from experience, ANNs have been mainly applied to data regression and classification problems. As a particular regression task, ANNs have been also widely used for time series prediction (Zhang et al., 1998; Crone, 2005; Zhang et al., 2001; Neural Forecasting Competition [Online]).

The rationale of the proposed method is to use ANNs as real-time neural predictors (RTNPs) in order to utilize the prediction errors to *both* continuously train the ANNs *and* to recognize the current motion using wearable sensors.

The latter ones are being increasingly used in many applications related to the monitoring and the recognition of daily living activities in healthy people (Pentland, 2005). Being these sensors easy to use and wear, they can be used in home environment allowing to implement tele-rehabilitation programs able to remotely monitor gesture performance (Schasfoort et al., 2002), and eventually to quantify the progress of rehabilitation in people recovering from pathologies. Thus, they are also being used to analyze and quantify human motion, for both lower limb and upper limb applications (Veltink et al., 1996; Mathie et al., 2004). Accelerometers are also often used in combination with magnetometers, cameras and gyroscopes to study tremor and balance, and for pose reconstruction (Giansanti et al., 2003). The related signals are analyzed by various processing methods, ranging from classic frequency analysis, to HMM-based techniques, to different types of Template Matching (Muscillo et al., 2007).

In this proposal, for each gesture a different RTNP is trained to predict each of the time series of the accelerometers channels, constituting a bank of RTNPs. After few consecutive motion repetitions, the recognition system guesses the gesture category referred to the RTNPs bank exhibiting the best predictions. The use of ANNs and of the on line learning scheme gives the system a powerful ability to learn different gestures and to be adaptive in learning the way different subjects perform the same gestures.

Although very challenging test conditions were considered (the RTNPs banks are not specialized to the related gesture, only the first gesture trains the ANNs, no signal processing is performed) an encouraging mean value of the percentage recognition rate greater than 83% is obtained, proving the robustness and the recognition power of the proposal, which will be detailed in the following

4.2. Materials and Methods

4.2.1. The proposed real-time neural predictor

ANNs are a neurologically inspired computational paradigm using many simple elaboration units (neurons) highly interconnected. A set of significant inputs and corresponding desired output couples (*training set*) is used to *train* the ANNs connections strengths (*weights*) minimizing the distance between the desired outputs and the actual outputs. According to the neurological long-term potentiation principle – the efficacy of synapses change as a result of experience providing both memory and learning to the brain – the training reinforces or depresses the connections giving the ANN the capability to learn the knowledge and the behavior contained in the training set. Thus, ANNs have been mainly applied to data regression and classification tasks.

A particular regression task is the time series prediction: L delayed samples {x[n-ℓ]; ℓ=0, 1,…, L-1} of the series are provided as L inputs to the ANN; the ANN H outputs give the predictions of the H future values {$\hat{x}[n+h]|n$; h=1, 2, …, H}. A non linear AR(L) model is thus assumed for the series (equation 3.1 (Zhang et al., 1998 and 2001; Crone, 2005).

$$x[n+1] = f(x[n-L+1],...,x[n-1],x[n]) \tag{3.1}$$

Assuming N previous samples of the series are available and considering a one-step-ahead prediction, the ANN training set is schematized in Table 4.1 (Zhang et al., 1998).

An essential feature of ANNs is the option to train them continuously even during functioning. The so called *on-line training* is conceptually opposite to the generally used *batch training* method, referred to the off-line ANN training.

Item No	Inputs	Desired output
1	x[1], x[2],…, x[L]	x[L+1]
2	x[2], x[3],…, x[L+1]	x[L+2]
…	…	…
N-L	x[N-L], x[N-L +1],…, x[N-1]	x[N]

TABLE 4.1 *Time series forecasting: generally used training set*

A basis of this proposal is to use ANNs as RTNPs. Referring to Figure 4.1, at each time *n* the RTNP stores the last N samples, predicts the H future values and uses the H prediction errors $e_h[n]$ (equation 3.2) to concurrently perform the training during the prediction phase, according to Table 4.1.

$$e_h[n] = x[n] - \hat{x}[n]|n-h \quad \mathrm{h} = 1, 2, ..., \mathrm{H} \tag{3.2}$$

These prediction errors will serve as inputs for the performance measurement step, described in the following.

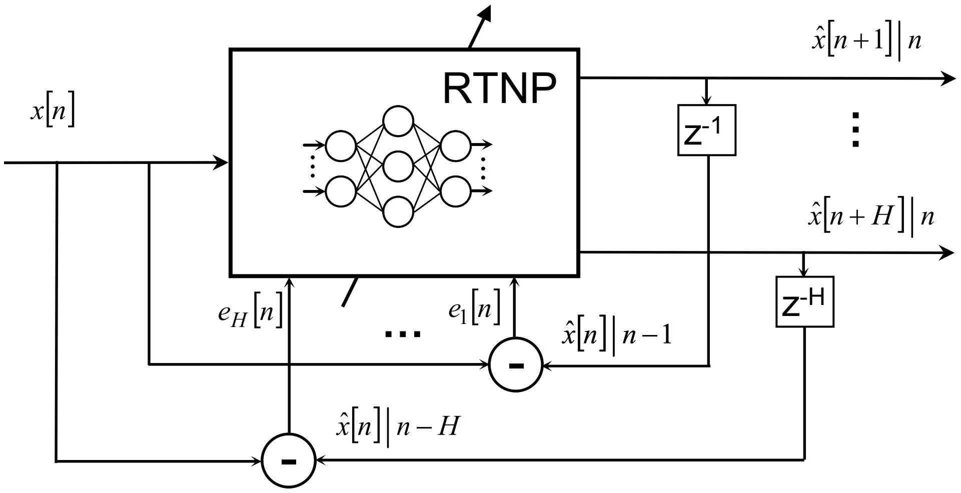


FIGURE 4.1 *Real-time neural predictor (RTNP)*

4.2.2. Performance measurements

Although the crucial performance measurement for predictors is the prediction accuracy, based on the prediction error (equation 3.2), a suitable figure for any given problem is not defined (Zhang et al., 1998; Neural Forecasting Competition [Online]). There are absolute performance measurements, which are used to compare the performances of different predictors operating on the same dataset:

- the mean square error (MSE) $= (\sum_n e^2[n])/N$
- the root mean square error (RMSE) $= \mathrm{MSE}^{1/2}$
- the mean absolute deviation (MAD) $= (\sum_n |e[n]|)/N$

and normalized performance measurements, which are in turn used to compare different data sets:

- the mean absolute percentage error (MAPE)

$$MAPE = \frac{100}{N} \cdot \Sigma_n \left| \frac{e[n]}{x[n]} \right| \tag{3.3}$$

- the symmetric mean absolute percentage error (SMAPE)

$$SMAPE = \frac{100}{N} \cdot \Sigma_n \frac{|x[n] - \hat{x}[n]|}{|x[n] + \hat{x}[n]|/2} \tag{3.4}$$

As in this proposal the several RTNPs prediction performances are compared to recognize gestures, a normalized performance measurement has to be used. Moreover, since the accelerometer signals used in this work have zero mean, the use of the MAPE would lead to very large values when $x[n]$ is close to zero. On the other hand the SMAPE is used for positively defined series. A corrected SMAPE (cSMAPE) (equation 3.5) is thus proposed.

$$cSMAPE = \frac{1}{N} \cdot \Sigma_n \frac{|x[n] - \hat{x}[n]|}{(|x[n]| + |\hat{x}[n]|)/2} \tag{3.5}$$

The performance measurement of the recognition system is the correct recognition percentage rate (CRPR).

4.2.3. Application: the accelerometers signals

The signals were recorded using two different accelerometers (ADXL202): a three-axis one, placed on the inside of the forearm (Acc1), and a dual-axis one placed on the upper arm (Acc2). The axes are oriented as in Figure 4.2.

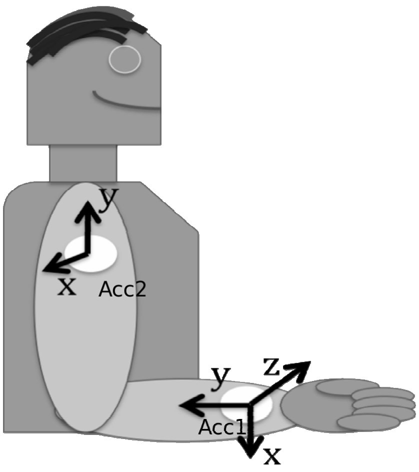


FIGURE 4.2 *Placement of the accelerometers*

For each subject 100 time series (5 repetitions × 5 channels × 4 gestures) were recorded. Signals were sampled at 400 samples/s and have a duration ranging from 2.5 s to 6.5 s (*raise pencil*, RP and *stack pieces*, SP task, respectively).

4.2.4. Application: experimental setup and protocol

Three healthy subjects were recruited to execute five repetitions of each exercise. The gesture categories are selected from a set of exercises listed in the Wolf motor function test (WMFT) which is generally used to monitor the improvement of functional movements in people recovering from stroke. The following four exercises are selected as they are more similar to the gestures performed during daily living:

- *SP - stack pieces* – stack up 3 different draughts pieces,
- *LD – lock the door* – reach, grasp and turn the keys as if locking a door three times,
- *RJ – raise jar* – reaching and raising a jar and bring it to the mouth, mimicking drinking,
- *RP – raise pencil* – reaching and then raising a pencil at a distance of 20 cm from the subject.

4.2.5. The proposed real-time gesture recognition system

In this section a real-time implementation of the use of RTNPs to recognize gestures is detailed.

A multilayer neural network (MNN) with one hidden layer is used for the RTNP. The standard error backpropagation (EB) training algorithm is used and the transfer functions for the hidden and the output layer units are the hyperbolic tangent (tanh) and the linear function, respectively.

The selection of the main MNNs factors is essentially problem-dependent and none of the existing methods can be assumed as superior to the others. Given the real-time feasibility requirement, an heuristic trial-and-error approach in modeling the RTNPs is so applied. For each gesture category a different bank of RTNPs is trained with the different accelerometer output channels (a RTNP for each channel). Four gesture categories are considered and five motion repetitions are totally available from the 3+2 channels of the two used accelerometers.

With reference to Figure 4.3, the proposed system is composed of 4 RTNPs banks, each being formed by 5 RTNPs. Only the single initial repetition of each gesture trains the RTNPs of the related bank and then the signals given by the 5 accelerometer channels are simultaneously provided to all the RTNPs banks so that H=40 series of the prediction errors $e_h[n]$ (equation 3.2) are evaluated.

After 4 movement repetitions of the same gesture, the mean cSMAPE for each RTNPs bank is evaluated averaging both in the 5 different channels and in the prediction directions h=20, 21,…, 40. The bank with the minimum mean cSMAPE provides the system guess about the gesture category.

The finally settled RTNPs model uses a training size of 5000 samples (the first repetition of each gesture, with N=1200 in Table 4.1), 600 input nodes, 2 hidden units and 40 output nodes.

As other authors found (Zhang et al., 1998; Crone, 2005; Zhang et al., 2001), it was confirmed that:

- the number of input nodes is the most critical factor for prediction and recognition task,
- a large training data set may overcome the overfitting problem,
- parsimonious models have both the best recognition performance and the highest generalization capability.

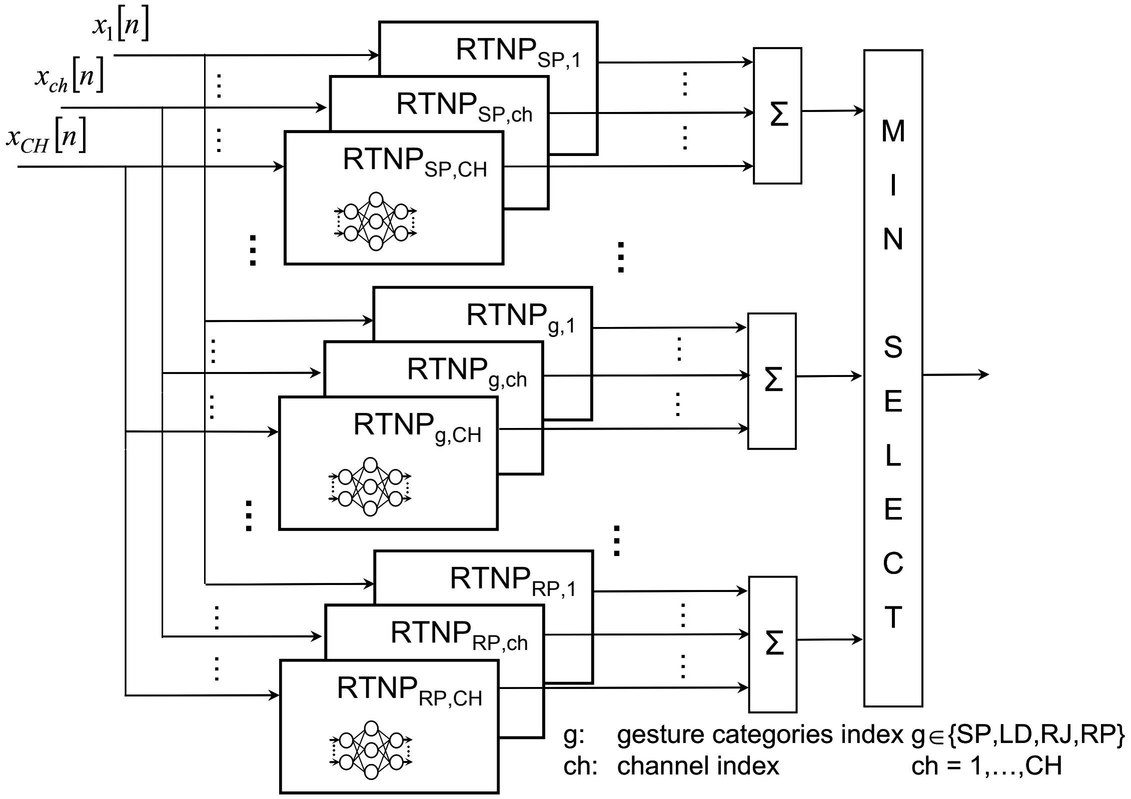


FIGURE 4.3 *Real-time gesture recognition system*

4.3. Results and Discussion

Figure 4.4 shows the predictions for the 5th accelerator channel of a SP gesture during the on-line training. The first repetition of each gesture is used twice during the training.

Figure 4.5 shows the cSMAPE for each of the 4 accelerometer channel RTNPs when RJ is performed: the minimum cSMAPE is found in the RTNP trained with RJ. Moreover, the higher is the prediction step, the higher is the cSMAPE but the higher is the system discrimination power.

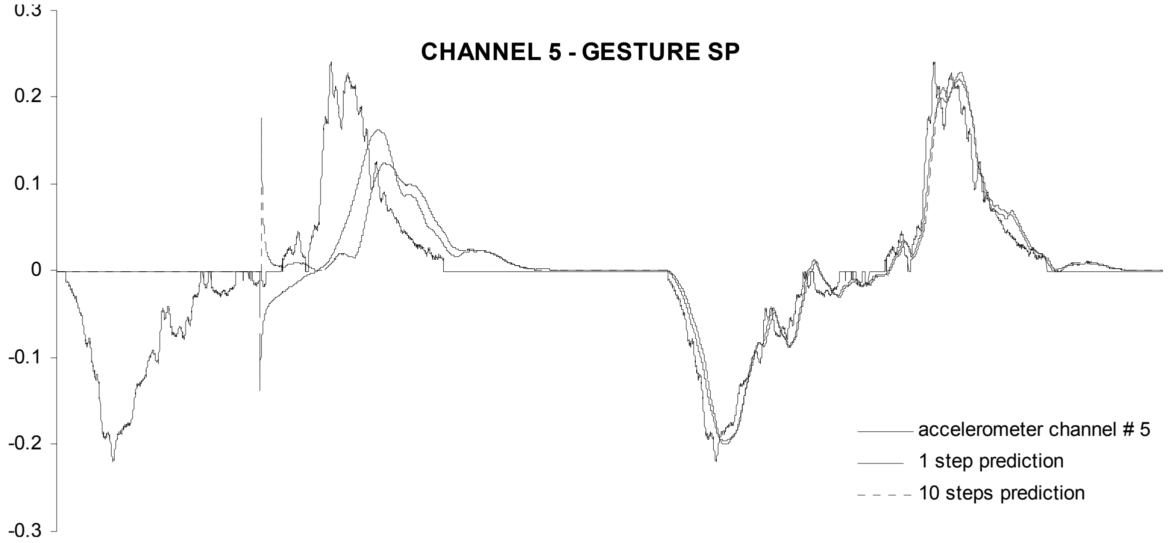


FIGURE 4.4 *Real time prediction during the on line training*

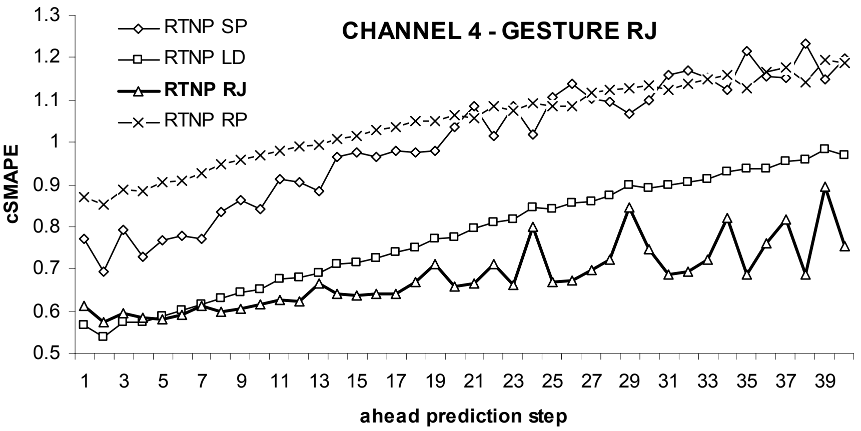


FIGURE 4.5 *Different gestures RTNPs - cSMAPE vs. prediction step*

It is also worth highlighting that:

- for each subject the RTNPs are re initialized; only the first gesture repetition is used for the training; the train is never again repeated,
- the different RTNP bank structure is not optimized for the specific related gesture category,
- for each bank the different cSMAPEs of the related RTNPs are simply averaged over the channels,
- neither any assumption nor any pre-processing or normalization of the data is

performed.

Even in these very challenging test settings, a mean CRPR higher than 83% is obtained, proving the robustness and the recognition power of the proposed system (Table 4.2).

Gesture / Subject	Gesture correct detection percentage rate				
	SP	LD	RJ	RP	Mean
Mean	100%	66.7%	66.7%	100%	83.3%

TABLE 4.2 *Time series forecasting: generally used training set*

4.4. Conclusions

An implementation of a real time gesture recognition system based on the use of time series neural predictors is proposed. The CRPR is measured on real data obtained from two inertial sensors placed on the upper limb. The accelerometer sensors, the ANNs architecture, and the prediction accuracy block are assembled in the system.

The use of ANNs and of the on line learning scheme gives the system a powerful ability to learn different gestures and to be adaptive in learning the way different subjects perform the same gestures.

As the learning can be repeated during the functioning, the system is also able to learn the change in performing movements for the same subject, suggesting applications in remote monitoring of gesture performance, and in adaptive rehabilitation pro-grams.

The lack of both a priori assumptions and signal pre processing makes the system prone to recognize different kind of gestures and, in general, of data patterns.

Even in the described test conditions, results obtained in this preliminary version of the system prove both the correctness and the robustness of the inspiring principles and the feasibility of a real time implementation.

To improve the recognition ability allowing the system to recognize motions at the end and even during the single gesture repetition, the following future works may be planned:

- to specialize each RTNPs bank architectures for the related specific gesture category,
- to sophisticate the ANNs (e.g. recurrent networks),

- to authomatize the ANNs modeling (e.g. generating more than a model and selecting the winning one (Crone, 2005)),
- to use more than one gesture repetition to train each RTNPs and periodically refreshing the training (e.g. using the trusted current recognition).

In order to overcome the problems due to failures in eye features detection and head motion, the neural time series prediction described in this Chapter may be included within the neural mapping function of the EGTS proposed in Chapter 2.

Chapter 5

New self-organizing meaningful artificial neural networks

ABSTRACT

The infinite support of sigmoidal activations of multilayer neural networks (MNNs) cause slow learning rate, lack of physical meaning, negative interference. This may prevent the useful application of ANN on eye-gaze tracking giving, in particular, slow calibrations. Localized receptive field (LRF) networks have promised similar regression power and faster learning than MNNs, and physically meaningful modeling. Unfortunately, LRF networks have often large size and/or performance worse than MNNs due to unsupervised placing and shaping of identical and radially symmetric kernels.

As networks of new LRF, called quadratic exponential elliptical neuron (QuEEN), can be reduced to opportune MNNs, the standard error backpropagation allows each QuEEN to be self placed and shaped by a supervised training. The separability of the hidden units of QuEEN networks allows parallelism and growing/pruning strategies. According to simulations, QuEEN networks showed comparable regression power and faster learning than MNNs, keeping the pros of LRFs and MNNs and overcoming the respective cons[*].

[*] Results described in this Chapter have been submitted for Publication in *Neuocomputing*.

5.1. Introduction

The success of neural networks (ANNs) – and in particular the standard multilayer neural networks (MNNs) – can be attributed to the ability of approximating any multivariate non-linear measurable function with an arbitrary degree of accuracy (Cybenko, 1989; Hornik, Stinchcombe, & White, 1989).

The success of a particular neural network model can be strongly connected to the properties of its learning algorithm, so that the huge diffusion of MNNs was certainly due to the computational efficiency and the quite ease of use of the error backpropagation (EB) algorithm and its variants (Rumelhart, Hinton, & Williams, 1986; Widrow & Lehr, 1990) (e.g. the virtual absence of any a priori assumption on the system to be modeled). Despite the huge set of application tasks ranging from the multivariate regression to the time series prediction (Zhang, Patuwo, & Hu, 1998), from the nonlinear control to the eye tracking systems (Gneo, Schmid, Conforto, & D'Alessio, 2012), some age-old drawbacks have raised some skepticisms on the application of MNNs. Among these we highlight those due to the infinite support of sigmoidal functions that are used as neuron activation rules: the slow learning rate (Cybenko, 1989; Hornik, Stinchcombe, & White, 1989), the lack of physical meaning of the representation built during the training (Rumelhart, Hinton, & Williams, 1986), the negative interference (Schaal & Atkeson, 1998), and the unfeasibility of the parallel implementation.

Network of localized receptive fields (LRFs), such as the well-known and popular radial basis functions (RBFs), would in principle overcome those drawbacks (Powell, 1987; Park & Sandberg, 1991; Park & Sandberg, 1993).

Unfortunately, the real concern with a RBF network (RBFN) is the optimization process for the choice of the RBF centers and smoothing factors (Chen, Cowan, & Grant, 1991): this choice is data dependent (Webb & Shannon, 1998), and suboptimal centers may reduce the learning rate (Broohmhead & Lowe, 1988). The usually adopted learning scheme performs the unsupervised LRF placing and shaping with no exploitation of the knowledge about the desired input-output mapping. This sub optimal two-phase scheme, together with the excessive simplification given by identical and radially symmetric LRFs (as the RBFs), often leads to networks with unreliable behavior, large size and performance worse than MNNs (Schwenker, Kestler, & Palm, 2001).

As neither the same smoothing factors, nor the radial symmetry for the LRFs are required to prove they are universal approximators (Park & Sandberg, 1991; Park & Sandberg, 1993), we derived the requirements an ideal LRF network should have:

1. it should be formed by elliptical and differently shaped LRFs,
2. all the network adjustable parameters should be jointly trained in a supervised procedure.

Similarly to what reported in (Lapedes & Farber, 1987; Poggio & Girosi, 1990a), we propose a multidimensional elliptical Gaussian LRF with different variances along each different input, so meeting the first requirement. The proposed network can be implemented as an opportune not fully connected two hidden layers MNN with quadratic and exponential activations. We therefore call its kernel a quadratic-exponential elliptical neuron (QuEEN). Moreover, since a standard EB can be applied to QuEEN networks to jointly train all the QuEEN's parameters (i.e. centers, smoothing factors and heights) the second requirement is also respected.

All the QuEEN units of the network self-organize by jointly placing, shaping and dimensioning themselves during the fully supervised EB training that exploits the whole knowledge about the desired input-output mapping included in the (input, desired-output) pairs of the training set, with no need of either a priori assumptions or unsupervised learning phases.

Contrarily to conventional RBFN – that cannot estimate the influence of each input on output – the physical meaning of QuEEN networks also allows to evaluate the importance factor, let's call it S, of each input variable on the output also derived in (Yeh, Zhang, Wu, & Huang, 2010a and 2010b), so that the model resulting after the training may provide an interpretation of the modeled system.

Regarding the two-phase learning usually performed for RBFNs, we highlight that neither the numerical complexity nor the time needed to determine the LRFs centers and smoothing factors have ever been taken into account to assess different ANN models. We therefore introduce a complexity measure given by the total number of the adjustable parameters of the network, so that a comparison between the conventional MNNs and the QuEEN networks may be performed in terms of regression power and learning rate.

Through numerical simulations, we will show that QuEEN networks exhibit a similar regression power with a considerably faster learning than MNNs, so confirming what was found about LRF in (Lapedes & Farber, 1987; Moody & Darken, 1988; Poggio & Girosi, 1990a).

Therefore the proposed QuEEN networks allow keeping the advantages of both MNNs (in terms of popularity and easiness of use of the EB algorithm and its variations) and LRF networks (in terms of fast learning rate and physical meaning), overcoming their respective drawbacks.

This work is structured as follows: in Section 5.2 a brief review about MNNs, LRFs and RBFs is provided, together with the analysis of their respective advantages and drawbacks. In Section 5.2.2 the proposed approach is introduced together with some considerations about the physical interpretation of QuEEN networks and the information that such representation may give on the modeled system. A complexity network measure is also introduced to allow comparing QuEENs with MNNs. In Section 5.4 the study of simple regression tasks and the comparison in terms of regression power and learning rate, between QuEENs and MNNs with the same complexity, is performed. In Section 5.5 the results and the conclusion of this work are summarized and some hints are provided about future work.

5.2. Background

5.2.1. Multilayer neural networks and backpropagation

The generalization power of ANNs is related with the ability to correctly predict the output values for inputs not contained in the training set. Learning and generalization are among the most useful attributes of ANNs (Widrow & Lehr, 1990).

When ANNs are applied to a regression task, the learning corresponds to finding a surface on the input space that gives the best fit to the training data following one optimum criterion, whereas generalization means interpolation between (and possibly extrapolation outside the range of) the sample data points along the regressing surface built during the training.

MNNs belong to the larger class of feedforward neural networks (FNNs), where the data processing flows from the input nodes towards the output, and the related graphs have no cycles.

Each unit of a MNN mimics the *all-or-none* behavior of biological neurons, which give a complete (and limited) response if stimulated above a certain activation threshold or, otherwise, give no response at all. This behavior is traditionally modeled by *squashing* sigmoidal functions quickly saturating as input moves away from a threshold towards negative and positive values (the *logistic* function monotonically grows assuming values in [0,1], whereas the hyperbolic tangent similarly ranges in [-1,+1]).

Since the single-hidden layer MNN class – the simplest nontrivial class of FNNs – with sigmoidal hidden units gives universal arbitrarily good approximators (Cybenko, 1989; Hornik, Stinchcombe, & White, 1989), we only consider one-hidden layer one-output MNNs (the extension to multidimensional output is obvious). We refer to this class of ANNs as conventional MNNs.

Conventional MNNs have been traditionally trained using the standard EB algorithm (Rumelhart, Hinton, & Williams, 1986; Widrow & Lehr, 1990; Brown, 1996), which implements the steepest descent rule to minimize the mean square error (MSE) surface in the weight space by iteratively scanning each pattern of the training set.

Notwithstanding their impressive diffusion, the following age-old claimed EB drawbacks have raised some skepticisms on the application of MNNs:

- the slow learning rate (Cybenko, 1989; Hornik, Stinchcombe, & White, 1989), and
- the lack of physical meaning of the representation of the modeled system built during the training (Rumelhart, Hinton, & Williams, 1986).

Despite its numerical efficiency due to recursion, EB is a first order steepest descent, which suffers from slow convergence due to the possibility of zigzagging about the actual direction to a local minimum. More sophisticated second order variants may generally improve convergence rates (e.g. *quasi-Newton* algorithm) (Broohmhead & Lowe, 1988), whereas some known factors and corrections may be applied to the weights update equation to modulate the learning factor and/or introduce a memory of the past weight updates via the so called *momentum* (Rumelhart, Hinton, & Williams, 1986; Widrow & Lehr, 1990). Notwithstanding that, the EB learning rate may be so slow to allow the application of MNNs to only “off-line” static tasks where training is performed once and for all, whereas the application to real-time adaptive systems, where repeated training may be required with strict time constraints, may be precluded (Moody & Darken, 1988).

Regarding the lack of physical meaning of the MNNs, during the learning an internal representation of the desired mapping regression task is built by the training of the network weights. The difficulty related to the physical meaning of this representation is due to its distributed nature, as it is the whole pattern of activity over all the hidden units – and not the meaning of any particular hidden unit – that is relevant. This is directly related to the non-limitation of the support of the sigmoidal activation functions (Rumelhart, Hinton, & Williams, 1986).

Among other secondary but not negligible drawbacks of MNNs, we also highlight:

- each MNN unit is activated on the basis of the input that is the sum of the outputs from all the preceding units; thus, a total parallelism is not realizable in MNN implementations,
- the so called *negative interference* effect (Schaal & Atkeson, 1998): due to the non-local nature of sigmoidal basis function, any change to the MNN weights has non-local effect that may lead to the effect known as negative interference when the input-output relationship is not stationary.

As any weight update during learning only greedily reduces the error related to the current training data, MNNs usually show excellent interpolation power but poor generalization capability outside of the range of training data. Any further training phase updates the MNN's weights to fit the new data, and may catastrophically lose the accuracy in the previous range of data.

In the following Section we highlight that, as the other described drawbacks, also the slow learning rate of MNNs is due to the infinite support of the sigmoidal activations. ANNs with compactly supported kernels are thus potentially able to overcome all the mentioned drawbacks of MNNs.

5.2.2. Localized receptive fields

In this work we informally – and with no distinction – refer to a *bump* or *localized receptive field* (LRF) as a function on an Euclidean space $\boldsymbol{R^n}$ having continuous derivatives of any order on a compact support. In this support it assumes values different from zero and has its maximum (the function *center*), whereas it has zero values or vanishes out of the support (Lapedes & Farber, 1987; Moody & Darken, 1988).

The interpretation of the Fourier analysis/synthesis as a superposition of half-period bumps (within suitable spectra of wavelengths), heuristically demonstrates that regression can be performed by a one hidden layer ANN whose hidden units are LRFs conveniently placed in the input space, with the coefficients of the regression given by the weights of a linear output unit.

The mapping performed by MNNs with sigmoidal units is interpretable as a summation of opportune bumps that the EB procedure needs to appropriately form, scale and place (Lapedes & Farber, 1987). Some gain in terms of learning rate can be so expected if the bumps of the network are available from the beginning instead of having the network build its own bumps from sigmoidal non compactly supported surfaces. Furthermore, only the small fraction of hidden LRFs centered very close to a given input are involved during both the network operation and training, whereas all sigmoidal units of a MNN must be trained for each input pattern (Moody & Darken, 1988; Moody & Darken, 1989).

Regarding the network physical meaning, unlike MNNs, the biological plausibility and the physical interpretation of LRF networks regard not only each single unit, but also the representation given by the whole units interconnection.

The network is formed by resonating hidden units each of which is a “receptive field” responding only for inputs falling in an input space region around its center, out of which its influence is rapidly vanishing. Locally-tuned neurons with “selective” response for some range of the input variables are found in many parts of the biological nervous systems. For example cochlear stereocilia (auditory system) have locally-tuned response to limited bands of frequencies, or cells in the visual cortex respond selectively to stimulation, which is both local in retinal position and local in certain directions of the visual field (Poggio & Girosi, 1990b). These locally-tuned neurons only respond to a limited and small range of the input space (Schwenker, Kestler, & Palm, 2001). This representation is believed to be important for improving signal to noise ratio and fault tolerance.

With the exception of the stereocilia cells, having locally tuned response given by their biophysical properties, locality is generally a property of the overall *system* and should not be confused with the biophysical response properties of cells, usually modeled in MNNs as a thresholding/squashing of a weighted sum of inputs. This historical knowledge regarding neurobiological locality of LRFs is fully reported and referenced in (Moody & Darken, 1988; Moody & Darken, 1989; Schaal & Atkeson, 1998).

Thanks to LRF locality, a spatially localized learning is possible, since the training data falling in some limited range maximally affect only the nearest LRFs, whereas distant LRFs are only negligibly affected (Schaal & Atkeson, 1998). Therefore, LRF networks allow parallel implementations and show natural robustness to negative interference.

5.2.3. Radial basis function networks

Most part of the literature regarding LRF networks has been related to radially symmetrical LRFs known as radial basis functions (RBFs), and RBF networks (RBFNs) are shown to offer an alternative to regression by MNNs with sigmoidal activations (Powell, 1987; Poggio & Girosi, 1990a; Poggio & Girosi, 1990b).

A RBF can be modeled by a scalar function Φ defined on $\boldsymbol{R^n}$, with a single maximum in its center $\boldsymbol{c}$, that vanishes as the (usually Euclidean) norm ||·|| of the vector difference between the input $\boldsymbol{x}$ and center $\boldsymbol{c}$ grows, accordingly to a scalar non negative smoothing factor σ also known as radius as in equation (4.1):

$$\Phi(\boldsymbol{x};\boldsymbol{c},\sigma) = \Phi(\|\boldsymbol{x}-\boldsymbol{c}\|;\sigma) \quad \boldsymbol{x},\boldsymbol{c} \in \Re^n, \sigma \in \Re^+ \tag{4.1}$$

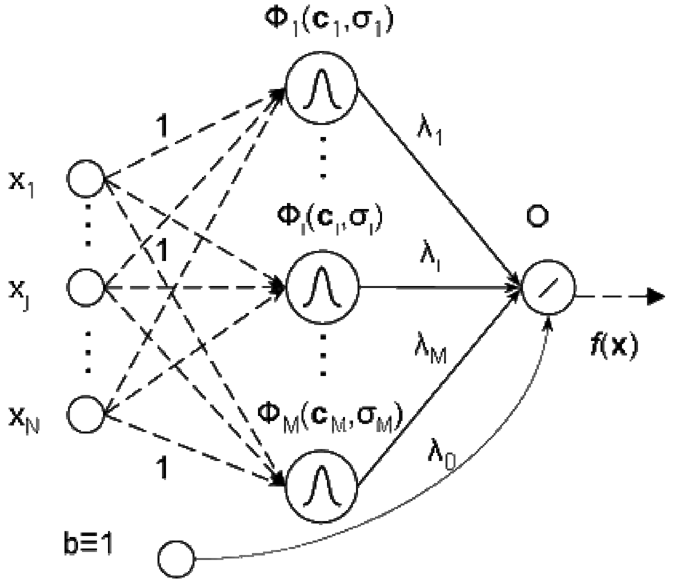
Similarly to MNNs, we consider a one-dimensional output space without loss of generality, so that the generic RBFN with M hidden units can be expressed as in equation (4.2).

$$f(\boldsymbol{x}) = \sum_{i=1}^{M} \lambda_i \Phi_i(\|\boldsymbol{x}-\boldsymbol{c}_i\|;\sigma_i) \tag{4.2}$$

If M input-output training pairs are available, each of the M RBFs is centered in a different input vector with a fixed radius, and the resulting function is constrained to go through the known M points. The weighting coefficients λ_i in equation (4.2) are uniquely given by the inversion of the M×M square matrix of the linear system (its non-singularity is proven for a large class of functions Φ if the data points are all distinct (Broomhead & Lowe, 1988)). When the number M' of data exceeds the number M of RBFs and the condition that RBF centers correspond to the data points is relaxed, the determination of the coefficients λ_i becomes an overspecified – but still linear and uniquely solvable – least squares optimization problem (e.g. via the Penrose pseudo-inverse, or single value decomposition).

Since a global shift of the desired function is hard to achieve by RBF weighting, an output bias weight λ_0 may be considered as provided by a trivial bias unit b having infinite radius and output fixed to 1. The mapping thus implemented is given by equation (4.3), and the corresponding RBFN is depicted in Figure 1, where each continuous arch is a trainable weight.

$$f(\boldsymbol{x}) = \lambda_0 + \sum_{i=1}^{M} \lambda_i \Phi_i(\|\boldsymbol{x}-\boldsymbol{c}_i\|;\sigma_i) \tag{4.3}$$



$\Phi_1(\mathbf{c}_1,\sigma_1)$
$\Phi_I(\mathbf{c}_I,\sigma_I)$
$\Phi_M(\mathbf{c}_M,\sigma_M)$
x_1
x_J
x_N
1
1
1
λ_1
λ_I
λ_M
λ_0
O
$f(\mathbf{x})$
b≡1

FIGURE 5.1 *A conventional radial basis function network (RBFN)*

We just highlight here that RBFN in Figure 5.2 is similar but *not* equivalent to an one-hidden layer MNN with RBFs as activations – with prefixed centers and radii – instead of sigmoids. Firstly each input is directly linked to each RBF by a unitary and fixed connection (dashed arches are not trainable and fixed to 1). Moreover, there are some other striking differences, better described in the following paragraph, also regarding the role of the bias unit and, above all, the hidden units. We will show as a LRF network similar to the RBFN in Figure 4.1 can be implemented by an opportune not fully connected *two*-hidden layer MNN.

The hugely claimed simplicity of the RBF training is due to the described theoretical non-iterative procedure to evaluate *only* the output weights λ_i, but this implies that all the RBF centers and radii have to be in some way pre-determined, whereas the real concern with RBFNs is the (best) choice of *appropriate* RBF centers *and* smoothing factors (Chen, Cowan, & Grant, 1991). This choice is data dependent (Webb & Shannon, 1998), and suboptimal centers may reduce the learning rate (Broohmhead & Lowe, 1988), while the choice of the RBF shape seems to be not crucial (Chen, Cowan, & Grant, 1991).

As RBFNs were mainly introduced to overcome the slow convergence of EB in MNNs, the simultaneous learning of all the network parameters performed by EB has been generally replaced by a *two-phase* learning that separates the evaluation of the RBF centers and smoothing factors of the hidden layer from the training of the output weights (Schwenker, Kestler, & Palm, 2001).

The literature has been so concerned with centers that a whole classification scheme for the different learning strategies derived for RBFNs positioning has been introduced (Karayiannis, 1997; Karayiannis, 1999), whereas the smoothing factors optimization has been almost ignored:

1. the RBFN has a RBF for each data included in the training set (the RBF centers are chosen to be training input vectors),
2. the RBF centers are randomly selected from the training input data,
3. the RBF centers are evaluated via the inputs clustering,
4. the RBF centers are evaluated via supervised procedures using the desired output information.

Although the first three unsupervised strategies do not consider at all the information about the desired output included in the training set, they are the most used to find out the RBF centers in the first phase of learning.

After the RBF centers evaluation, the second phase learns just the output weights (i.e. the RBF heights) via the mentioned least mean squares rule.

Notwithstanding its large use, the two-phase learning scheme was recognized as ineffective and responsible for RBFNs having unreliable behavior, large size or performance that is worse than MNNs (Schwenker, Kestler, & Palm, 2001).

First of all, the separation of the learning in two phases cannot obviously lead to RBFN implementing models that are optimal in a global way (Webb & Shannon, 1998; Yeh, Zhang, Wu, & Huang, 2010a; Yeh, Zhang, Wu, & Huang, 2010b).

Secondly, the second phase can be ill-conditioned when the RBF centers are too close to each other (the application of singular value decomposition is then required) (Chen, Cowan, & Grant, 1991).

Moreover, the selection of RBF centers given by the first two unsupervised strategies is arbitrary and clearly unsatisfactory (Chen, Cowan, & Grant, 1991). The use of as many RBFs as input known patterns is actually reasonable only for “encoding” problems (e.g. the exclusive-OR) where the limited and noise-free training set is given by the dictionary of the code and there is no need to generalize inputs not belonging to that dictionary (Broohmhead & Lowe, 1988). The selection of RBF centers is the simplest alternative to the previous strategy when there is plenty of training data. In this case no suitable selection strategy is easily applicable if the function to be regressed is not known, and the centers are thus often chosen randomly. The unsupervised clustering of the input vectors (e.g. *k*-means clustering

(Moody & Darken, 1988; Moody & Darken, 1989) or linear vector quantization (Karayiannis, 1997; Karayiannis, 1999)), puts similar (near) input patterns in the same cluster and set the RBF centers as the centroids of the clusters. Clustering algorithms like *k*-means are heuristics whose results may strongly depend both on the number and on the choice of the initial clusters, and some problems may arise when the input patterns are not equally distributed (Chiu, Cook, Pignatiello, & Whittaker, 1997).

Furthermore, the two-phase learning scheme generally chooses identical RBF smoothing factors without any particular strategy, and kept them fixed during training as not persuasive procedures have been developed and relatively little effort has been made to optimize RBF smoothing factors. At best, they are evaluated in an intermediate phase of the two-phase learning scheme, between the centers determination and the output weights calculation, by means of unsupervised contiguity heuristics that try to find a tradeoff between RBF locality and smoothness (i.e. overlapping).

Besides the mentioned reasons of trouble, accordingly to other authors we believe that the main problem with the two-phase learning is that the information about the desired mapping given by the input-output pairs in the training set is usually ignored during the first unsupervised phase of centers evaluation (Karayiannis, 1999). Moreover, also the described smoothing factors evaluation, when performed, is unsupervised (Guillén, Rojas, González, Pomares, Herrera, Valenzuela, & Rojas, 2007). The application of unsupervised strategies to determine RBF centers (and smoothing factors) to regression tasks represented by two different training sets having the same input vectors (e.g. given by the same input sampling scheme) but different output targets, will come out with two RBFNs having the *same* hidden layer: this simple example outlines the criticality of this topic.

Moreover, the target function can have a large bandwidth in some areas (i.e. high variability) where a lot of (narrow) RBFs may be needed, and narrow bandwidth in other areas (i.e. smooth behavior), where just a few (wide) RBFs are needed. Thus, the unsupervised center evaluation could unsuitably place RBFs where the target function is smooth or easy to model, whereas the naïve use of a high number of narrow RBFs densely placed in the input space, simply gives non-practicable RBFNs.

We thus strongly also believe that RBF smoothing factors should be different, allowing the radii to be different in different areas (Guillén, Rojas, González, Pomares, Herrera, Valenzuela, & Rojas, 2007). As a matter of fact, even if RBFNs with the same radius can still be universal approximators (Park & Sandberg, 1991; Park & Sandberg, 1993), RBFs with identical radii should be avoided (Benoudjit & Verleysen, 2003) as when each RBF can define its own value for the radius, the RBFN performance can be increased, whereas inadequate values can severely impair the regression power. Different radii can be evaluated by an unsupervised input clustering, considering the centroids and the standard deviations of the training data in each cluster, but although this approach is far better than the fixed-width methods, the smoothing factors so evaluated still remain sub-optimal (Benoudjit & Verleysen, 2003).

Another important criticism of the RBFNs regards the radial symmetry of traditional RBFs: the regression so performed is not invariant to scaling of the input variables unlike MNNs (Webb & Shannon, 1998), and the circular contour lines of RBFs implicitly consider that all the input variables have equal weights. Therefore, conventional RBFNs cannot estimate the influence of inputs on the output, so that the resulting model is unsuitable to provide an interpretation of the modeled system.

As *neither* the same smoothing factors, *nor* the radial symmetry for LRFs are required for the proof of the universal approximation capability (Park & Sandberg, 1991; Park & Sandberg, 1993), and the influence of each independent variable on the desired mapping has a generally different scale, a more reasonable LRFs should have *oval* contour lines (Yeh, Zhang, Wu, & Huang, 2010a; Yeh, Zhang, Wu, & Huang, 2010b).

Summarizing all the previous considerations about RBFNs, we thus conclude that networks of LRFs with the following characteristics should be considered:

1. all the LRF adjustable parameters (smoothing factors, centers and output weights) have to be *jointly* optimized during a fully supervised training, taking into account the output values of the target function, and
2. each kernel function needs N different smoothing factors for the N different input variables, so to obtain not symmetrical LRFs.

These two conditions have been practically never considered together. A gradient descent based supervised learning minimizing the mean quadratic RBFN output error was developed in (Karayiannis, 1997; Karayiannis, 1999; Karayiannis & Randolph-Gips, 2003), but smoothing factors are prefixed, whereas only the centers and output weights are trained.

A partially supervised approach was proposed where the number of RBFs is iteratively increased starting from one and continuing by adding RBFs chosen from a large set of candidates so that the explained variance of the desired output is maximized (Chen, Cowan, & Grant, 1991). Sadly, the candidate RBFs correspond to centers still chosen from the data points, and the smoothing factors optimization is not considered at all. In reference (Chiu, Cook, Pignatiello, & Whittaker, 1997), different smoothing factors for the different LRFs were trained together with output weights in a supervised learning based on gradient descent, but the centers are evaluated, as usual, via input unsupervised clustering. In reference (Benoudjit & Verleysen, 2003), the performance deterioration due to RBFs with identical scalar and fixed widths is described and the smoothing factor optimization is considered in the case of symmetrical RBFs. Shape-adaptive RBFs were proposed and two learning schemes were derived in (Webb & Shannon, 1998), the first with radially symmetrical RBFs, each with a different scalar smoothing factor, and the second with identically shaped standard symmetrical RBFs. Shape-adaptive RBF networks give rise to lower errors and smaller networks than conventional RBFNs, but the centers are still evaluated via the suboptimal unsupervised *k*-means clustering of the inputs. A supervised approach where RBF centers and smoothing factors are trained so that RBFs are placed where the target function is highly variable, whereas centers are rarefied in smooth evaluated areas is derived in (Guillén, Rojas, González, Pomares, Herrera, Valenzuela, & Rojas, 2007), but symmetrical RBFs are considered. As smoothing factors are evaluated after centers, and then are the output weights, a joint optimization was not performed.

In the following section we propose a network of powerful *elliptical* multivariate Gaussian LRFs, each having a multidimensional smoothing factor. We also show that such a network can be implemented by a convenient MNN so that the standard EB allows *jointly* training all the network adjustable parameters during a fully supervised training scheme, whereas the speed of the training is kept so to combine the advantages of both MNNs and LRFs, overcoming their drawbacks.

Networks of adaptive radial basis function (ARBF) having oval contour lines were recently proposed (Yeh, Zhang, Wu, & Huang, 2010a; Yeh, Zhang, Wu, & Huang, 2010b). Conventional symmetrical RBFs were fed with weighted input variables and a supervised steepest descent learning scheme was derived for RBFs centers, radii, output and input weights. ARBF networks resulted much more accurate than RBFNs, but as the input weights indirectly give the elliptical shape of the kernel functions, the real need to increase the network complexity to adaptively train also the RBF radii is not clear.

An efficient fully supervised learning scheme of networks having generalized multivariate Gaussian kernels, referenced as hyper basis functions (HBFs), was proposed in (Poggio & Girosi, 1990a) with performance comparable to MNNs. The contours of HBFs with a generic covariance matrix are hyperellipsoids in the input space that have arbitrary orientation (Schwenker, Kestler, & Palm, 2001).

The network kernel proposed in this work (QuEEN) can be considered as a separable HBF having diagonal covariance matrix, with the axes of (hyper)elliptical contour lines parallel to the axes of the input space. As the proposed approach considers QuEEN networks as particular MNNs, it allows the *direct* application of *just* the standard EB.

5.3. Quadratic Exponential Elliptical Neurons Networks

In this work we propose a LRF given by a multidimensional Gaussian function *Q* having diagonal covariance matrix with different smoothing factors along the different input variables, gathered in the vector $\boldsymbol{\sigma}$ as in equation (4.4), where $\mathbf{\Sigma}$ is the inverse of the covariance matrix with the diagonal given by the square of the component of the vector $\boldsymbol{\sigma}$, and T denotes the transpose of a matrix.

$$\begin{aligned} &Q(\boldsymbol{x};\boldsymbol{c},\boldsymbol{\sigma}) = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{c})^T \Sigma(\boldsymbol{x}-\boldsymbol{c})\right), \\ &\boldsymbol{x},\boldsymbol{c},\boldsymbol{\sigma} \in \Re^N,\ \Sigma = diag\left(\sigma_1^{-2},\ldots,\sigma_N^{-2}\right),\ \boldsymbol{\sigma} = [\sigma_{\boldsymbol{1}},\ldots,\sigma_{\boldsymbol{N}}]^T \end{aligned} \qquad (4.4)$$

Given the diagonality of the matrix $\mathbf{\Sigma}$, the equation (4.4) is separable and so is the kernel function *Q* so that the Eq.(4) can be expressed as in equation (4.5)

$$Q(\boldsymbol{x};\boldsymbol{c},\boldsymbol{\sigma}) = \mathbf{exp}\left(-\left(\frac{(x_1-c_1)^2}{2\sigma_1^2} + \ldots + \frac{(x_N-c_N)^2}{2\sigma_N^2}\right)\right) \qquad (4.5)$$

We thus can define N basic functions, the *j*-th of which $q_j(x_j;c_j)$ simply gives the square of the projection on the *j*-th input space dimension of the difference between the input $\boldsymbol{x}$ and the center $\boldsymbol{c}$ vectors, the kernel function *Q* may also be expressed as in Eq.(6), where the negative exponential is defined as *E()* for simplicity.

$$\begin{aligned} &Q(\boldsymbol{x};\boldsymbol{c},\boldsymbol{\sigma}) = E\left(\sum_{j=1}^{N}\frac{q_j(x_j;c_j)}{2\sigma_j^2}\right), \\ &q_j(x_j;c_j) = (x_j-c_j)^2,\ E(x) = \mathbf{exp}(-x),\ \mathrm{j} = 1,\ldots,\mathrm{N} \end{aligned} \qquad (4.6)$$

The *Q* kernel unit given by equation (4.6) can be so implemented by an elementary MNN given by the connection of N basic units with quadratic activation to an output basic unit having the negative exponential *E(x)* as activation, where N is the input space dimension. The *j*-th quadratic unit thus implements the function $q_j(x_j;c_j)$, and receives as input the *j*-th variable of the input space and the *j*-th output of a trivial bias unit, whose N output weights implement the (opposite of the) components of the *Q* center ***c***.

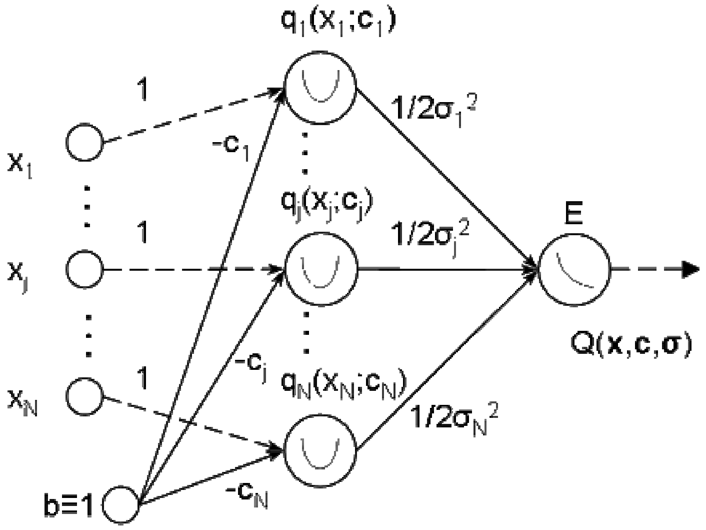


FIGURE 5.3 *A quadratic-exponential elliptical neuron (QuEEN)*

In Figure 5.3 we show the implementation of the QuEEN already defined as a multidimensional Gaussian unit. The possibility to freely update both the QuEEN center and its smoothing factors is represented by continuous arches, whereas the dashed arches indicate one valued connections from the input

$$f(\boldsymbol{x}) = \lambda_0 + \sum_{i=1}^{M} \lambda_i Q_i(\boldsymbol{x};\boldsymbol{c}_i,\boldsymbol{\sigma}_i) \tag{4.7}$$

Similarly to equation (4.3), in equation (4.7) the regression performed by the superposition of M QuEENs is expressed, whereas the related network of QuEENs conveniently placed, shaped and weighted is shown in Figure 5.4.

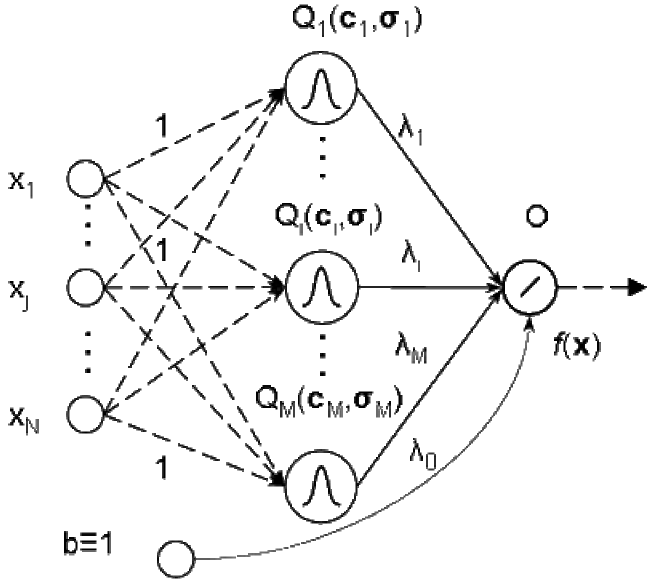


FIGURE 5.4 *A quadratic-exponential elliptical neuron (QuEEN) network*

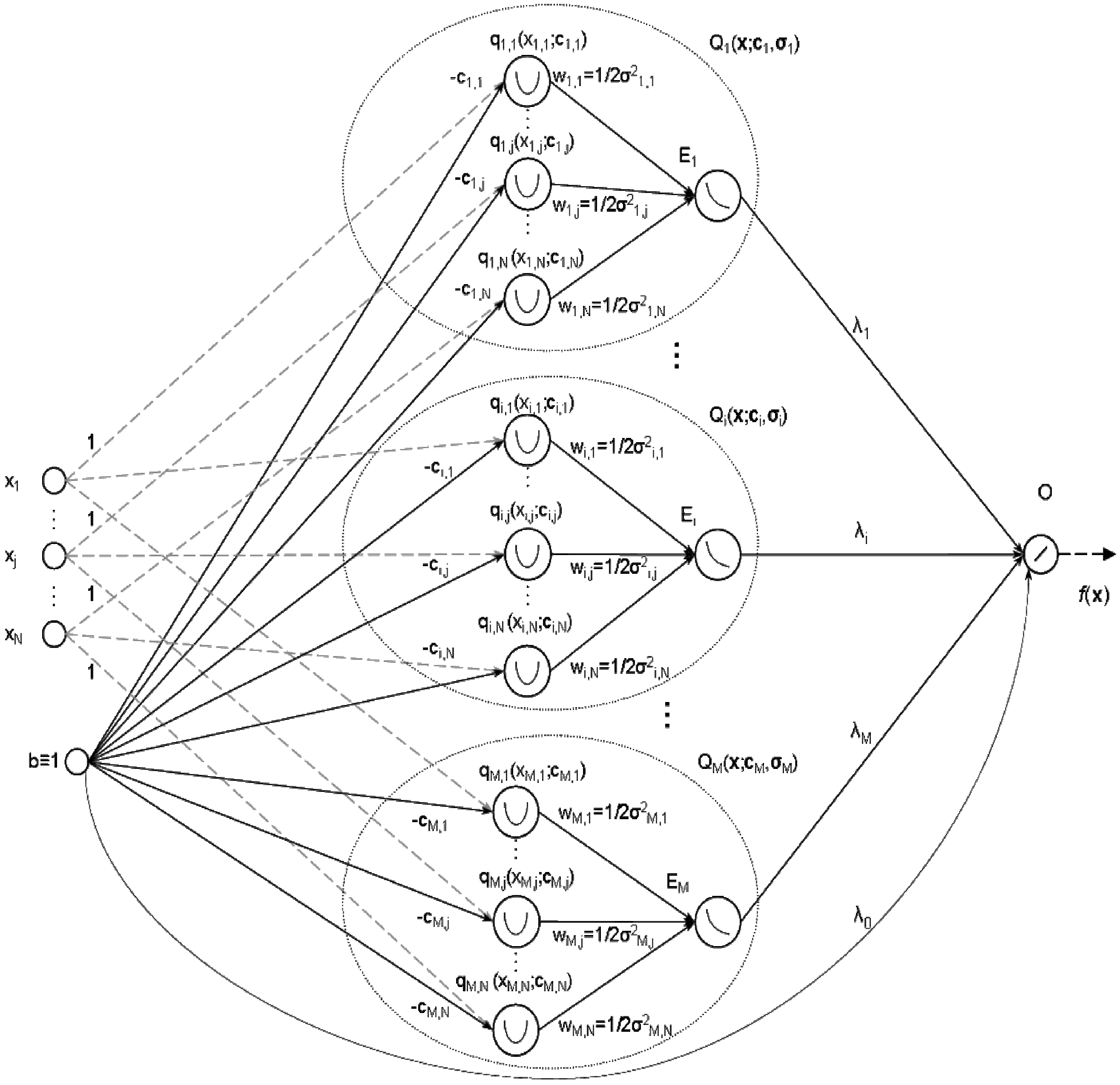
The striking differences between the QuEEN network in Figure 5.4 and a one-hidden layer MNN, whose units have multidimensional Gaussian activations instead of sigmoids, are:

- *input layer connections*: MNNs input connections need to be trained, whereas QuEEN networks have unitary fixed input connections,
- *role of the bias unit*: in MNNs the bias unit is connected to all the hidden and output units, whereas the QuEEN bias is just connected to the output unit,
- *input of the hidden units*: the hidden units of a MNN apply the activation function to the weighted sum of all the incoming inputs, whereas each QuEEN applies the conveniently placed and shaped elliptical activation just to the input,
- *parameters of hidden units*: all the hidden units of a MNN have the same shape and no adjustable parameters (even the unit threshold is externally given by the bias connection), whereas a QuEEN network needs the output weights training and the centers and smoothing factors of each QuEEN.

If in each term in equation (4.7) every QuEEN function is expressed as in equation (4.6), the following equation (4.8) is straightforwardly obtained:

$$f(\boldsymbol{x}) = \lambda_0 + \sum\nolimits_{i=1}^{M} \lambda_i \underbrace{E\left(\sum_{j=1}^{N} \frac{q_{i,j}\left(x_{i,j};c_{i,j}\right)}{2\sigma_{i,j}^2}\right)}_{Q_i(x;c_i,\sigma_i)} \qquad (4.8)$$

The equation (4.8) expresses the particular MNN depicted in Figure 5.5 – where N is the input space dimension and M is the number of regressing units – that can be also graphically derived if each unit in the QuEEN network in Figure 5.4 is implemented in terms of quadratic and exponential unit as in Figure 5.3.



$q_{1,1}(x_{1,1};c_{1,1})$
$Q_1(\mathbf{x};\mathbf{c}_1,\mathbf{\sigma}_1)$
$-c_{1,1}$
$w_{1,1}=1/2\sigma^2_{1,1}$
$q_{1,j}(x_{1,j};c_{1,j})$
E_1
$-c_{1,j}$
$w_{1,j}=1/2\sigma^2_{1,j}$
$q_{1,N}(x_{1,N};c_{1,N})$
$-c_{1,N}$
$w_{1,N}=1/2\sigma^2_{1,N}$
λ_1
$q_{i,1}(x_{i,1};c_{i,1})$
$Q_i(\mathbf{x};\mathbf{c}_i,\mathbf{\sigma}_i)$
1
$w_{i,1}=1/2\sigma^2_{i,1}$
x_1
$-c_{i,1}$
O
1
$q_{i,j}(x_{i,j};c_{i,j})$
E_i
λ_i
x_j
$-c_{i,j}$
$w_{i,j}=1/2\sigma^2_{i,j}$
1
$f(\mathbf{x})$
$q_{i,N}(x_{i,N};c_{i,N})$
x_N
$-c_{i,N}$
1
$w_{i,N}=1/2\sigma^2_{i,N}$
λ_M
b≡1
$q_{M,1}(x_{M,1};c_{M,1})$
$Q_M(\mathbf{x};\mathbf{c}_M,\mathbf{\sigma}_M)$
$-c_{M,1}$
$w_{M,1}=1/2\sigma^2_{M,1}$
$q_{M,j}(x_{M,j};c_{M,j})$
E_M
$-c_{M,j}$
λ_0
$w_{M,j}=1/2\sigma^2_{M,j}$
$q_{M,N}(x_{M,N};c_{M,N})$
$-c_{M,N}$
$w_{M,N}=1/2\sigma^2_{M,N}$

FIGURE 5.5 *Reduction of a QuEEN network to a not fully connected MNN*

The special MNN so derived, that we simply refer to in the following as a QuEEN network, exhibits the following distinctive features with respect to a conventional MNN:

- it is a two hidden layer MNN, whose first hidden layer is formed by M×N units having quadratic activations, and the second hidden layer is given by N units having a negative exponential activation,

- the hidden layer units are separable in M *isolated* groups (see dashed ovals in Figure 5.5), so that each group implements a QuEEN (see Figure 5.3) and is formed by N quadratic units and one exponential unit,
- given the disjunction of the previous M groups, the hidden layer units are *not* fully connected each other,
- even the N input clampers are *not* fully connected with the first hidden layer units as the *j*-th input variable is connected only to the *j*-th unit of each unit group,
- the connections between the input and the first hidden layer are *not* trainable as they have to be kept fixed to one, and
- the bias is connected only to the first hidden layer and to the output unit, whereas in conventional MNNs the bias unit is fully connected with all the units.

The EB algorithm strictly requires the network to be only a FNN whose all units have continuous and differentiable activations, whereas the units are *not* required either to be fully connected, or to show the same sigmoidal activation (Rumelhart, Hinton, & Williams, 1986). It follows that a fully supervised and *joint* learning of all the adjustable parameters of the QuEEN network (centers, asymmetric smoothing factors and heights) can be simply given by the standard EB.

5.3.1. Physical meaning of the QuEEN networks weights

As all the QuEEN network weights have an immediate physical meaning, differently from conventional MNNs, the whole QuEEN network representation built during the training has a direct physical interpretation.

With reference to all the trainable connections in the QuEEN network shown in Figure 5.5, it follows that:

- the *(i,j)*-th connection $-c_{i,j}$ from the bias unit to the first hidden layer units directly represents the (opposite of the) *j*-th input coordinate of the *i*-th QuEEN center,
- the *(i,j)*-th connection $w_{i,j}$ from the first to the second hidden layer is inversely proportional to the variance of the *i*-th QuEEN along the *j*-th input direction,
- the *i*-th output weight λ_i directly gives the height of the *i*-th QuEEN,

- the bias weight λ_0 gives the output function offset.

Thanks also to the separability in M groups of the hidden units, the evolution of the parameters of each QuEEN during the training is physically meaningful.

Contrarily to conventional RBFN – that cannot estimate the influence of each input on output, so that the resulting model is unsuitable to provide an interpretation of the modeled system (Yeh, Zhang, Wu, & Huang, 2010a and 2010b) – the proposed QuEEN networks are also prone to the following physical interpretation:

- the smaller is the *(i,j)*-th weight $w_{i,j}$, the larger is the smoothing factor of the *i*-th QuEEN along the *j*-th input direction. The vanishing of $w_{i,j}$ thus represents a scarce influence of the *j*-th input variable on the *i*-th QuEEN as it will give a constant value output along the *j*-th input direction,
- the vanishing of the *i*-th output weight λ_i (i.e. the height of the *i*-th QuEEN) represents a negligible response of the network for inputs around the center $\boldsymbol{c_i}=[c_{i,1}\ldots c_{i,j}\ldots c_{i,N}]^{\mathrm{T}}$ of the *i*-th QuEEN.

These qualitative considerations may find an analytical confirmation on the importance factor S_j of the *j*-th input variables on the output of a fully supervised adaptive radial basis function network recently derived through the first differential of the output on the *j*-th input in (Yeh, Zhang, Wu, & Huang, 2010b). If applied to a QuEEN network, the importance factor S_j gives the following equation (4.9):

$$S_j \equiv \sum_{m=1}^{M} abs\left(\frac{\lambda_m}{\sigma_{m,j}^2}\right) \tag{4.9}$$

where λ_m and $\sigma_{m,j}$ are respectively the output weights and the variance along the *j*-th input direction related to the *m*-th QuEEN. The bigger S_k as compared to S_j, the more the *k*-th input affects the network output with respect to the *j*-th input.

As the weights of a QuEEN network represent QuEEN centers and asymmetric smoothing factors, the EB training of QuEEN may be regarded as supervised clustering of the input space that takes into account the knowledge on the input-output mapping, and provides the centroids and the shape of M elliptical clusters.

5.3.2. A complexity measure for neural networks

What has been generally done in the assessment of ANNs is the quite rough performance comparison for different kinds of networks having the same number of hidden units, e.g. in terms of regression power and learning rate.

Notwithstanding this, given the same number of hidden units, different kinds of ANNs will generally have a different number of adjustable parameters.

We think that it may not have much sense, for example, to observe that, for a network having a certain number of adjustable parameters, the regression error converges faster than for another kind of network having many more adjustable parameters; it is indeed reasonable to expect that the more complex the network – given the same training set – the higher the number of learning cycles to converge.

In this work we aim to compare MNNs to QuEEN networks evaluating commensurable performances, and we choose to compare networks having exactly the same number of adjustable parameters to be jointly trained with the same standard EB.

Therefore, as each kind of ANN constitutes a C-parameter family of models where C is the total number of adjustable parameters which are varied during the training, we simply decided to consider the network complexity as given by the total number of adjustable parameters.

As a conventional MNN is fully connected and all of its connections have generally to be trained, its complexity is simply given by the total number of its connections.

Given N the input dimension and M the number of hidden units of the MNN, as the bias unit is connected to all the hidden and output units, the complexity of a generic one-layer (one output) MNN is given by equation (4.10).

$$C_{MNN} = (N+1)M + (M+1) \qquad (4.10)$$

As the RBFN learning has usually been performed with the two-phase learning scheme where the determination of RBF centers and smoothing factors is separated by the output weights training, the complexity of LRF networks has been roughly considered given by the number of hidden units.

Notwithstanding this, as also the LRF smoothing factors and centers are adjustable network parameters as the output weights are (i.e. the LRF heights), the QuEEN network complexity is given by all of its adjustable parameters as in equation (4.11).

$$C_{QuEEN} = 2MN + (M+1) \tag{4.11}$$

5.4. Test protocol and experimentation data

In the following we firstly regress Gaussian functions. These simple tasks allow validating the proposed EB application and help better understand the behavior of QuEEN networks, by analyzing the trajectories of the QuEEN network weights, thanks to their physical meaning.

Then we compare conventional hidden layer MNNs with sigmoidal units to the proposed QuEEN networks in terms of both regression power and learning rate. The comparison is performed for networks having the same complexity as defined in the previous section.

In this work we just consider the regression of bivariate functions as it allows obtaining graphical views of the results of the regression task, to better understand the QuEEN behavior without any loss of generality.

As in reference (Schaal & Atkeson, 1998), we consider the approximation of a two-dimensional desired function $f_d(x,y)$ whose samples are corrupted by (0 mean, 0.01 variance) additive Gaussian noise (AGN), whereas the training set is formed by 500 pairs of *(x,y)* input points randomly drawn from the square [-1.0,1.0]×[-1.0,1.0] of $\boldsymbol{R}^2$, and the related desired outputs are given by 500 noisy samples of the *z(x, y)* in equation (4.12). The test set is formed by 1681 data points taken on a 41×41 grid over the square [-1.0,1.0]×[-1.0,1.0] uniformly sampled in each dimension at 0.05 wide step.

$$z(x,y) = f_d(x,y) + N(0, 0.01) \tag{4.12}$$

The regression power is evaluated through the output root mean square error (RMSE) given by the difference between the actual network output and the $f_d(x,y)$ function value, evaluated on both the training and the test sets.

The learning time is qualitatively given by the number of epochs the network RMSE needs to converge to a quasi-constant value.

It is straightforward to see that, for a two dimensional input, the complexities of MNNs and QuEEN networks are given by equation (4.10) where N=2 and M gives the number of hidden units (i.e. QuEENs) respectively as reported in equation (4.13).

$$
\begin{aligned}
C_{MNN} &= 4M + 1 \\
C_{QuEEN} &= 5M + 1
\end{aligned}
\tag{4.13}
$$

In the following Table 4.1 the network architectures giving exactly the same complexity for two-input networks as functions of the number of hidden and QuEEN units are listed.

QuEEN Units no.	MNN Units no.	Network complexity
4	5	21
8	10	41
12	15	61
16	20	81
20	25	101
24	30	121
28	35	141

TABLE 5.1 *Complexity of a two input network in function of the number of hidden units*

The standard EB training is applied to both MNNs and QuEEN networks and the same learning factor and momentum are used. For both the tested architectures, the units of the output layer are linear, to avoid issues related to the output range. Both the MNN and the QuEEN network weights are randomly initialized. Three different realizations of random weights were considered for each network architecture. Each network realization was then trained with a different randomly formed training set. The RMSE values obtained were averaged over the three realizations for both QuEENs and MNNs.

5.4.1. Regression of one Gaussian function

We start our analysis considering the elliptical Gaussian bivariate desired function centered in the origin with standard deviations σ_x=0.2 and σ_y=0.6 as expressed in equation (4.14), and we regress it by one-QuEEN networks. This trivial regression task validates the proposed EB application and helps better understand the behavior of QuEEN networks, by analyzing the trajectories of the weights thanks to their physical meaning.

$$f_d(x,y) = \mathbf{exp}\left(-\left(\frac{x^2}{2(0.2)^2} + \frac{y^2}{2(0.6)^2}\right)\right) \tag{4.14}$$

The 3D surface diagram and the 2D contour diagram of such desired function are shown in Figure 5.6 a).

The regressing one-QuEEN two-input network is derived from equation (4.5), equation (4.8) and Figure 5.5 where M=1 and N=2, so that equation (4.15) and Figure 5.6 b) can be obtained.

$$f(x,y) = \lambda_0 + \lambda_i\,\mathbf{exp}\left(-\left(\frac{(x-c_x)^2}{2\sigma_x^2} + \frac{(y-c_y)^2}{2\sigma_y^2}\right)\right) \tag{4.15}$$

The network in Figure 5.6 b) was so randomly initialized and then trained using EB for 10000 iterations (epochs) of the training set built as previously described.

The trajectory and the evolution of each coordinate of the QuEEN center for one realization of the network are depicted in Figure 5.6 c) and Figure 5.6 d). Just after the first 100 epochs the QuEEN center quickly moves quite near the center of the desired function (it is only 0.06 far from the origin). The QuEEN then slowly approaches the origin reaching it after about 5000 epochs, whereas the center estimation oscillates around the correct value in the last 5000 remaining epochs (see the little dashed circle in Figure 5.6 c)).

The QuEEN center coordinate c_x convergence is faster than c_y as the target function steepness along the X coordinate is greater than the Y coordinate ($\sigma_y > \sigma_x$).

A similar trend is exhibited in Figure 5.6 e) by the output weights and the smoothing factors that quickly jump near their final values to which they slowly converge in less than 5500 epochs (only the first significant 6000 epochs are shown):

- the output offset λ_0 vanishes,
- the QuEEN smoothing factors respectively reach the target function standard deviations (σ_x=0.2, σ_y=0.6),
- the QuEEN height λ_1 reaches the correct value which is 1.

The RMSE related to the regression error was measured on both the training and the test set and is depicted in Figure 5.6 f). After a quick reduction, in about 5000 epochs both the errors approach the standard deviation 0.1 of the AGN so proving that the QuEEN network reaches a very good fitting of the target function.

The rightness of the proposed approach and the physical significance of the QuEEN network weights were so verified.

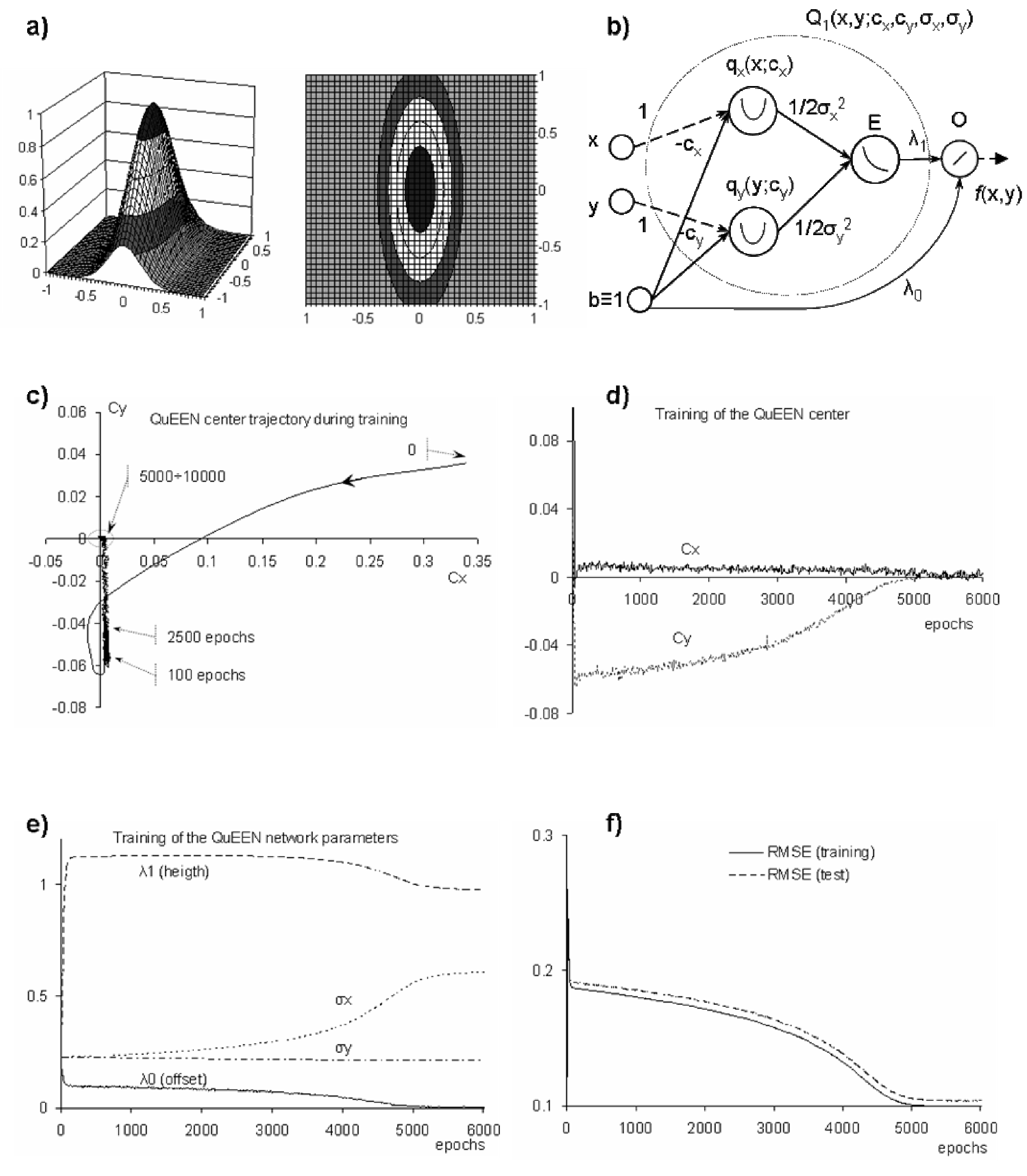


FIGURE 5.6 *a) 3D surface diagram and 2D contour diagram of a bivariate Gaussian function centered in the origin with σ_x=0.2 and σ_y=0.6.*

b) The one QuEEN two input regressing network.

c) Trajectory of the QuEEN center during the training.

d) Evolution of the QuEEN center coordinates during the training.

e) Evolution of output offset, smoothing factors and height of the QuEEN during the training.

f) Root mean square error (RMSE) evaluated on both the training and test set.

5.4.2. Regression of two Gaussian functions

Now we consider a bimodal desired function given by the superposition of two elliptical Gaussian bells and we regress it by a two-QuEEN network. The target function is expressed in equation (4.16), where the first bell is negative, centered in (-0.5,0.5) with standard deviations (0.6,0.2), and the second bell is positive, centered in (0.5,-0.5) with standard deviation (0.2,0.6). The related 3D surface and 2D contour diagrams are shown in Figure 5.7 a).

$$\begin{aligned} f_d(x,y) = -\mathbf{exp}\left(-\left(\frac{(x+0.5)^2}{2(0.6)^2}+\frac{(y-0.5)^2}{2(0.2)^2}\right)\right)+ \\ \mathbf{exp}\left(-\left(\frac{(x-0.5)^2}{2(0.2)^2}+\frac{(y+0.5)^2}{2(0.6)^2}\right)\right) \end{aligned} \tag{4.16}$$

The regressing two-QuEEN two-input network derived from equation (4.5), equation (4.8) and Figure 5.5 where M=2 and N=2, so that equation (4.17) and Figure 5.7 b) can be obtained.

$$\begin{aligned} f(x,y) = \lambda_0 + \lambda_1\,\mathbf{exp}\left(-\left(\frac{(x-c_{1,x})^2}{2\sigma_{1,x}^2}+\frac{(y-c_{1,y})^2}{2\sigma_{1,y}^2}\right)\right) \\ +\lambda_2\,\mathbf{exp}\left(-\left(\frac{(x-c_{2,x})^2}{2\sigma_{2,x}^2}+\frac{(y-c_{2,y})^2}{2\sigma_{2,y}^2}\right)\right) \end{aligned} \tag{4.17}$$

Three random realizations of the QuEEN network in Figure 5.7 b) were trained using EB for 50000 epochs of training sets built as previously described.

The trajectories and the evolution of each coordinate of the two QuEEN centers for one realization of the network are depicted in Figure 5.7 c) and Figure 5.7 d). Just after the first 100 epochs the two QuEEN centers quickly move quite near the maximum and minimum of the target function, whereas the center estimations oscillates around the correct values in the remaining epochs (see the little dashed ellipses in Figure 5.7 c).

Similarly to what already observed, the convergence of the first QuEEN center coordinate $c_{1,y}$ is faster than $c_{1,x}$ as the first Gaussian bell steepness along the Y coordinate is higher than the X coordinate ($\sigma_{1,x} > \sigma_{1,y}$). Similarly, the convergence of the second QuEEN center coordinate $c_{2,x}$ is faster than $c_{2,y}$ as the second Gaussian bell steepness along the X coordinate is greater than the Y coordinate ($\sigma_{2,y} > \sigma_{2,x}$).

A similar trend is exhibited in Figure 5.7 e) by the output weights and the smoothing factors that quickly jump near their final values to which they slowly converge in less than 15000 epochs (only the first significant 20000 epochs are shown):

- the output offset λ_0 vanishes very quickly,
- all the QuEEN smoothing factors respectively reach the local standard deviations of the target function,
- both the QuEEN heights λ_1 and λ_2 reach the minimum and the maximum values of the target function.

The RMSE related to the regression error was measured on both the training and the test set and is depicted in Figure 5.7 f). After a quick reduction, in about 10000 epochs both the errors approach the standard deviation 0.1 of the AGN so proving that, also in this case, the QuEEN networks reach a very good fitting of the target function.

As already found, the QuEEN parameters converge faster for the input coordinates along which the target function steepness is higher.

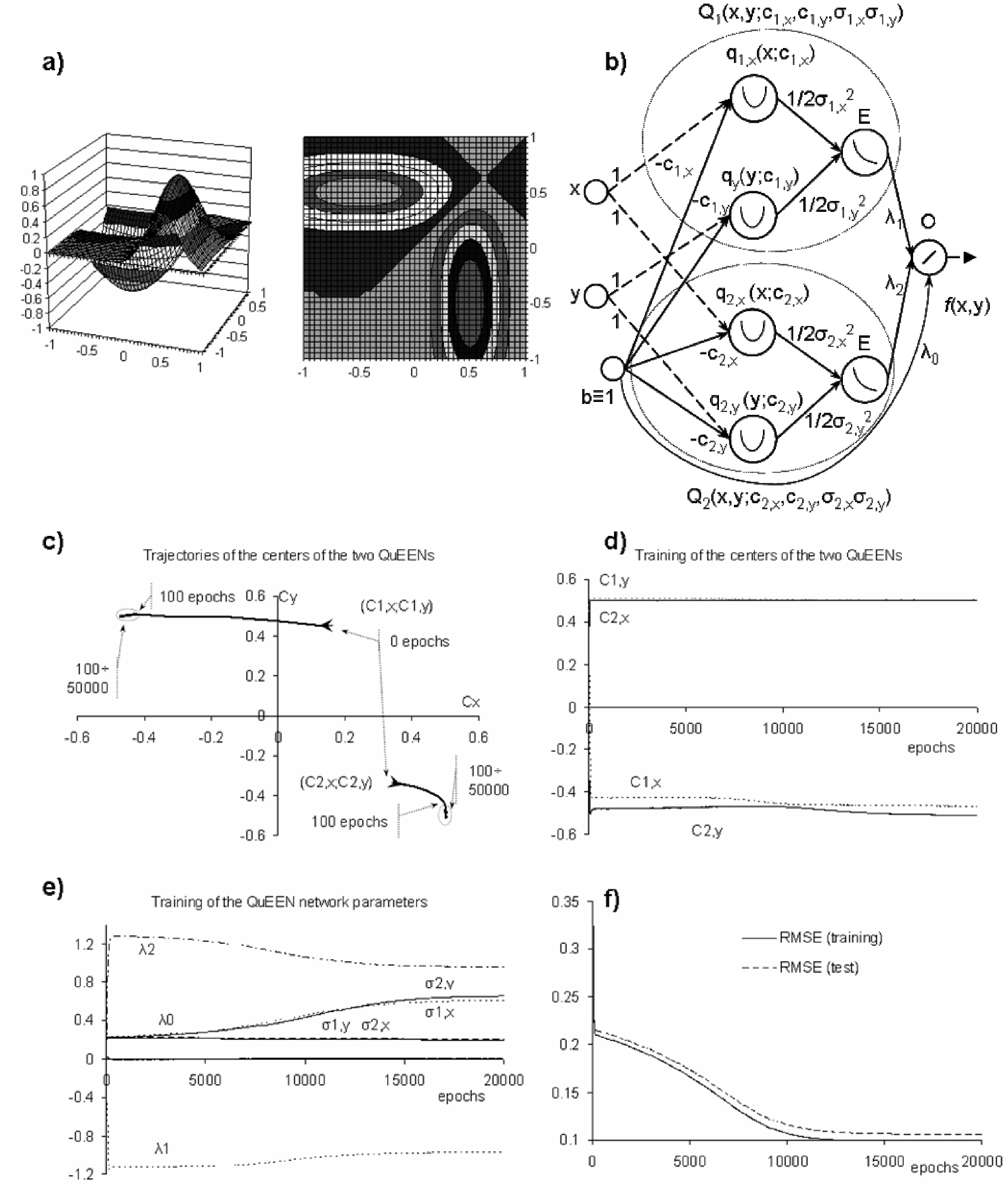


FIGURE 5.7 *a) 3D surface diagram and 2D contour diagram of a bimodal Gaussian target function.*
b) The two QuEEN two input regressing network.
c) Trajectory of the centers of the two QuEENs during the training.
d) Evolution of the two QuEENs center coordinates during the training.
e) Evolution of output offset, smoothing factors and height of the two QuEENs during training.
f) Root mean square error (RMSE) evaluated on both the training and test set.

5.4.3. Regression of bump functions

From now on we compare QuEEN networks to conventional MNNs with the same complexity in terms of regression power and learning rate for different target functions. We firstly consider a basic bump function as a target different from the trivial Gaussian bumps we used before. Afterwards, a more complex bump target function given by the superposition of basic bumps opportunely placed and scaled is introduced.

The first one is the basic bump function defined in equation (4.18), whose 3D surface and 2D contour diagrams are shown in Figure 5.8 a).

$$B_{ump}(x,y) = \begin{cases} \mathbf{exp}\left(-\left(1 - \|\boldsymbol{r}\|^2\right)^{-1} \right) & \|\boldsymbol{r}\| < 1 \\ 0 & otherwise \end{cases} \qquad (4.18)$$
$$\boldsymbol{r} = \begin{bmatrix} x & y \end{bmatrix}^T$$

Three random realizations of each network for both QuEENs and MNNs were trained using EB for 100000 epochs of training sets randomly built as previously described.

The RMSEs obtained for one-QuEEN networks and one, two and three sigmoidal hidden units MNNs were evaluated for both the training and test sets and respectively showed in Figure 5.8 c) and Figure 5.8 d).

MNNs with either one or two hidden sigmoidal units are not able to regress the basic bump function, whereas at least three hidden units are needed to reach the AGN, while a simply one-QuEEN network does.

This result is confirmed by the exam of the best regression given by respectively one, two and three sigmoidal hidden units MNNs after 100000 epochs, as depicted in Figure 5.8 b).

Then, at least three sigmoidal units are needed by the MNNs to build a bump during the training phase.

Similarly to what found in (Lapedes & Farber, 1987) for LRF networks, the one-QuEEN network is advantaged in the regression of such a bump target functions with respect than MNN that needs more complex configuration to build a bump during the training and reach the same accuracy.

This is confirmed by the evaluation of the network complexities as defined in Section 5.3.2: a MNN with three units has complexity 13, whereas an one-QuEEN network showing the same regression power has complexity 6.

The comparison between QuEEN networks and MNNs was then performed for networks having exactly the same minimum complexity that is given by 4-unit QuEEN networks and 5-unit MNNs, having complexity 21 (see Table 5.1).

The RMSEs evaluated for both the training and test set are respectively showed in Figure 5.8 e) and Figure 5.8 f) (only the first 2000 significant epochs are considered).

These results show that for the basic bump target function, QuEENs networks exhibit regression power similar to MNNs but with faster convergence as the AGN level is already reached after less than 500 epochs, whereas MNNs needs more than 1000 epochs.

In Figure 5.9 the regression of the target basic bump (Figure 5.9 a) performed by one realization of a 4-units QuEEN network is showed, together with the one performed by a 5-units MNN (same 21 complexity) after 100, 1000, 10000, and 100000 epochs.

After only 100 epochs the 4 QuEENs already grouped together to form a bump (Figure 5.9 b)), whereas the 5-units MNN is far from the bump shape (Figure 5.9 b')).

After 1000 epochs the 4 QuEENs are placed to form a better bump, whereas the global offset is closer to zero level than after 100 epochs (Figure 5.9 c)). The 5-units MNN have formed the bump, but the zero level offset is not correctly approximated, as the actual function goes down to negative values at the border of the domain (Figure 5.9 c')).

The QuEEN network bump approximation practically keeps its shape after 1000, 10000, and 100000 epochs (Figure 5.9 c), d), and e)), whereas the MNN only slightly improves its bump shape and the negative values at the domain border. epochs (Figure 5.9 c'), d'), and e')).

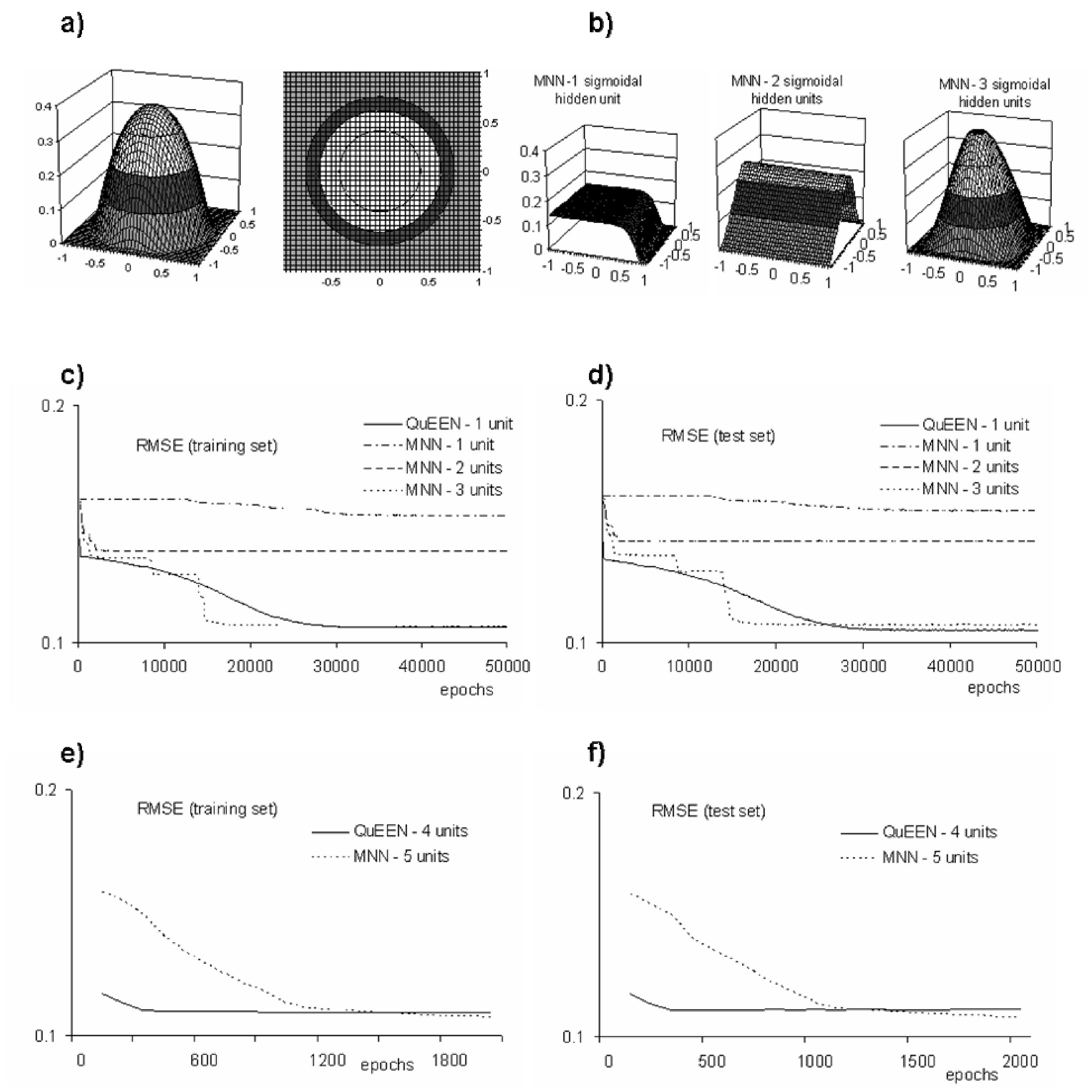


FIGURE 5.8 *a) 3D surface diagram and 2D contour diagram of the basic bump target function.*
b) Regression after 100000 epochs with MNNs having 1, 2 and t3 sigmoidal hidden units.
c) Regression by one QuEEN networks and 1, 2, and 3 sigmoidal hidden units MNNs: RMSE (training set).
d) Regression by one QuEEN networks and 1, 2, and 3 sigmoidal hidden units MNNs: RMSE (test set).
e) Regression by 4 units QuEEN network and 5 units MNNs: RMSE (training set).
f) Regression by 4 units QuEEN network and 5 units MNNs: RMSE (test set).

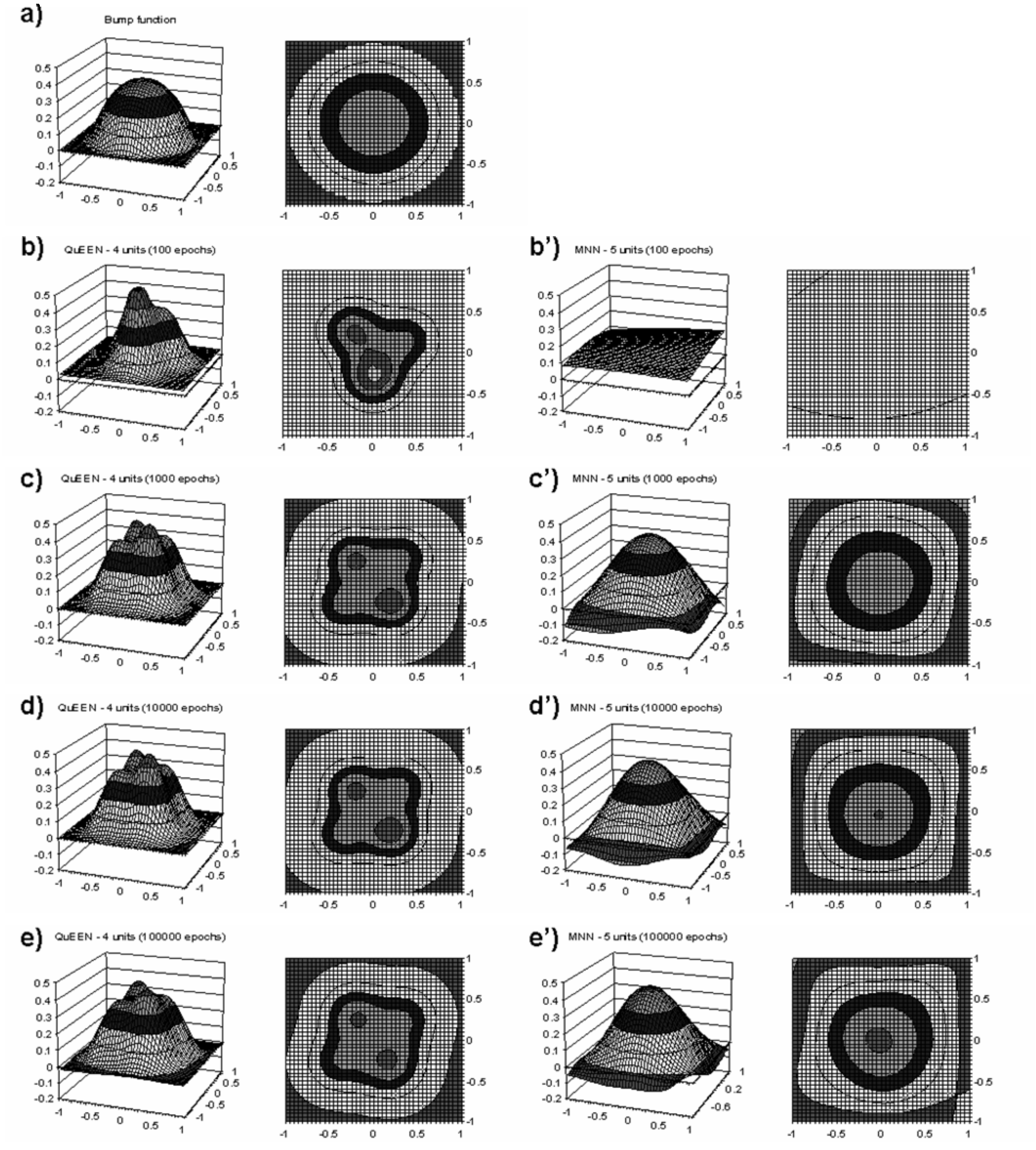


FIGURE 5.9 *Regression of the basic bump function by 5 units MNN and 4 units QuEEN network (complexity 21):*
a) 3D surface diagram and 2D contour diagram of the theoretical function.
b), b') Regression after 100 epochs.
c), c') Regression after 1000 epochs.
d), d') Regression after 10000 epochs.
e), e') Regression after 100000 epochs.

Then we considered a target function given by the superposition of seven basic bumps placed and scaled as in equation (4.19). Given its shape (the 3D surface and 2D contour diagrams are shown in Figure 5.10 a), we call it the *crown function*.

$$\begin{aligned}
C_{rown}(x,y) = B_{ump}(2x,2(y-0.5)) + \\
B_{ump}(2(x-0.375),2(y-0.325)) + \\
B_{ump}(2(x+0.375),2(y-0.325)) \\
B_{ump}(2(x-0.6),2y) + \\
B_{ump}(2(x+0.6),2y) + \\
B_{ump}(2(x-0.45),2(y+0.35)) + \\
B_{ump}(2(x+0.45),2(y+0.35))
\end{aligned} \tag{4.19}$$

Three random realizations of each network for both QuEENs and MNNs were trained using EB for 100000 epochs of training sets randomly built as previously described.

The RMSEs obtained for QuEEN networks and MNNs with the same complexity ranging from 21 (4-unit QuEENs, 5-unit MNNs) to 121 (24-unit QuEENs, 30-unit MNNs, see Table 5.1) were evaluated for both the training and test sets. Similarly to what found for the basic bump, the RMSEs on the training and test set are practically the same (see Figure 5.10 e) and Figure 5.10 f)). We thus only show in Figure 5.10 e) the RMSE on the test set after 100, 1000, 10000, and 100000 epochs of training.

Whereas, the higher is the number of epochs, the considerably lower is the RMSE of MNNs, the QuEEN networks show a faster learning rate (after only 100 epochs the QuEEN RMSE nearly approach the AGN) and a lower dependency on the number of epochs. The RMSE of QuEEN networks is also uniformly lower than MNN with the same complexity for all the considered learning time. Whereas MNNs show a RMSE slightly decreasing as the network complexity increase, the RMSE of QuEEN networks is practically not depending on the network complexity for complexities greater than 21. The higher is the number of epochs of training, the more marked is that behavior for both the networks.

The uniformly better regression power and the faster convergence of QuEEN network with respect to MNNs can be also verified from the exam of the RMSE decay during the training shown in Figure 5.10 f) for networks having complexity 21 and 121 (only the first significant 30000 epochs are showed). The higher the networks complexity, the faster the learning rate expressed in number of epochs, and the lesser the dependence of RMSE of QuEEN networks on the number of training epochs.

We conclude that when the regression of bump-like target functions has to be performed, provided the same complexity, the QuEEN networks show a generally better regression power with a considerably faster learning time than MNNs.

In Figure 5.10 the regression of the target crown bump (Figure 5.10 a)) performed by one realization of a 24-units QuEEN network is shown, both with the one obtained by a 30-units MNN (same 121 complexity) after 100, 1000, and 10000 epochs, respectively in Figure 5.10 b)-b'), c)-c'), and d)-d').

After only 100 epochs the QuEENs already formed a good approximation of the crown with the global offset already near zero (Figure 5.10 b)), whereas the MNN is very far from the crown shape (Figure 5.10 b')).

After 1000 epochs the QuEEN network shapes the crown slightly better (Figure 5.10 c)), whereas the MNN forms a bump still far from the crown shape, while negative values at the border of the domain are reached (Figure 5.10 c')).

After 10000 epochs the QuEEN network approximation is practically not changed (Figure 5.10 d)), whereas the MNN forms a bimodal bump still far from the crown shape with more negative values at the border of the domain (Figure 5.10 d')).

Summarizing the results, we found that MNNs need at least three hidden sigmoidal units to slowly form a bump during the training. This result corroborates what previously obtained showing that the regression of bump-like functions may be efficiently performed by QuEEN networks having less complexity than MNNs, whereas, given the same complexity, QuEEN networks exhibit similar or better asymptotic regression power and a faster learning rate than MNNs.

Furthermore, the negative interference due to the infinite support of the sigmoidal activation of MNNs is verified as the accuracy of MNNs is generally made worse by the non zero values assumed out of the target function support.

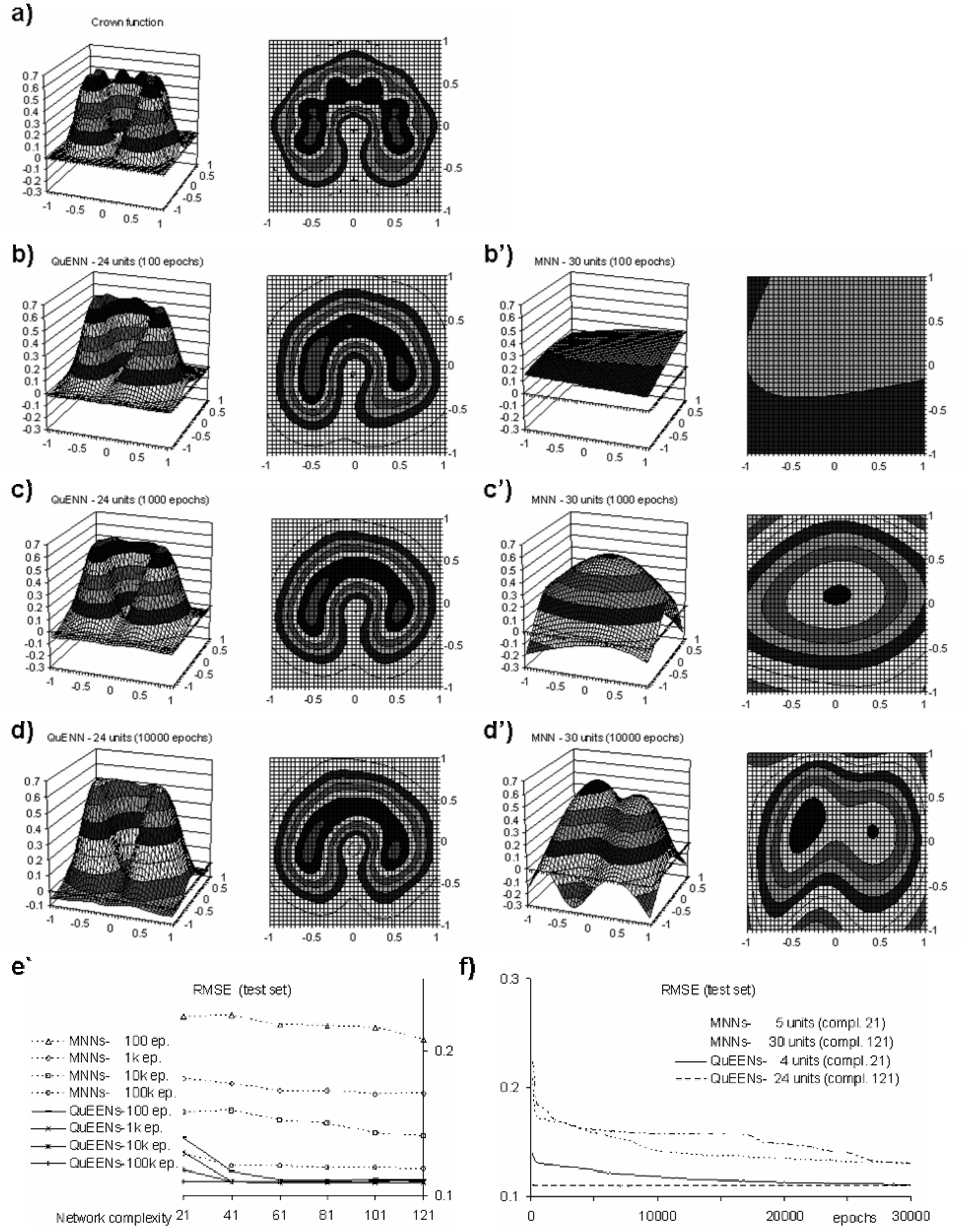


FIGURE 5.10 *Regression of the crown function by 30 units MNN and 24 units QuEEN network (compl. 121):*
a) 3D surface diagram and 2D contour diagram of the theoretical function.
b), b') Regression after 100 epochs, c), c') Regression after 1000 epochs.
d), d') Regression after 10000 epochs.
e) RMSE of MNN and QuEEN networks (compl. from 21 to 121) after different epochs (test set).
f) RMSE of MNN and QuEEN networks having complexity 21 and 121 (test set).

5.4.4. Regression of a well-known benchmarking function

Finally we compare QuEEN networks to conventional MNNs considering a well known sufficiently complex learning task given by the regression of the *crossed ridge* function (Schaal & Atkeson, 1998) expressed in equation (4.20), and whose 3D surface and 2D contour diagrams are shown in Figure 5.11 a).

$$C_{rossed-ridge}(x,y) = \mathbf{max}\left[\mathbf{exp}\left(-10x^2\right), \mathbf{exp}\left(-50y^2\right), 1.25\,\mathbf{exp}\left(-5\left(x^2+y^2\right)\right)\right] \tag{4.20}$$

Crossed ridge function if formed composing a Gaussian bump at the origin with a narrow and a wide ridge, which are perpendicular to each other.

Three random realizations of each network for both QuEENs and MNNs were trained using EB for 100000 epochs of training sets randomly built as previously described.

The RMSEs obtained for QuEEN networks and MNNs with the same complexity ranging from 61 (12-unit QuEENs, 15-unit MNNs) to 141 (28-unit QuEENs, 35-unit MNNs, see Table 5.1) were evaluated for both the training and test set.

As observed before, the RMSE on both the training and test sets is practically the same, we only show in Figure 5.11 e) the RMSE on the test set after 100, 1000, 10000, and 100000 epochs of training.

Whereas the higher the number of epochs, the considerably lower the RMSE of MNNs, the QuEEN networks show a faster learning rate (after only 100 epochs the QuEEN RMSE nearly approaches the AGN) and a lower dependency on the number of epochs. The RMSE of QuEEN networks is also uniformly lower than MNN with the same complexity for all the considered learning time with the only exception of 100000 epochs, after which the RMSE of MNNs is slightly lower than RMSE of QuEEN networks.

After 100 epochs MNNs show a RMSE decreasing as the network complexity increases with the exception of the highest considered complexity of 141, whereas both MNNs and QuEEN networks show a RMSE practically not dependent on the network complexity. The greater the number of epochs of training, the more marked that behavior for both the networks.

The similar regression power and the faster convergence of QuEEN network as compared to MNNs can be also verified from the exam of the RMSE decay during the training shown in Figure 5.11 f) for networks having complexity 61 and 141 (only the first significant 10000 epochs are showed). The higher the networks complexity, the faster the learning rate expressed in number of epochs, and the lesser the dependency of the QuEEN networks RMSE on the number of training epochs.

We conclude that when the regression of the crossed ridge function has to be performed, and the same complexity is considered, the QuEEN networks show a generally similar asymptotic regression power with a considerably faster learning rate than MNNs.

In Figure 5.11 the regression of the target crossed ridge function (Figure 5.11 a) performed by one realization of a 28-units QuEEN network is shown, together with the one obtained with a 35-units MNN (same 141 complexity) after 100, 1000, and 10000 epochs, respectively in Figure 5.11 b)-b'), c)-c'), and d)-d').

After only 100 epochs the QuEENs already formed a quite good approximation of the crossed ridge function with the global offset already near to zero (Figure 5.11 b)), whereas the MNN is very far from the crossed ridge shape (Figure 5.11 b')).

After 1000 epochs the QuEEN network better shaped the crossed ridge (Figure 5.11 c)), whereas the MNN formed a central bump still far from the crossed ridge shape while negative values at the border of the domain are reached (Figure 5.11 c')).

After 10000 epochs the QuEEN network slightly better shaped the crossed ridge (Figure 5.11 d)), whereas the MNN formed a shape quite resembling to crossed ridge but less accurate than QuEEN (Figure 5.11 d')).

Summarizing the results, we found that, given the same network complexity, QuEEN networks exhibit similar asymptotic regression power and a faster learning rate than MNNs.

Furthermore, the negative interference due to the infinite support of the sigmoidal activation of MNNs was verified as the accuracy of MNNs regression is generally made worse by the non zero values assumed out of the support of the crossed ridge target function.

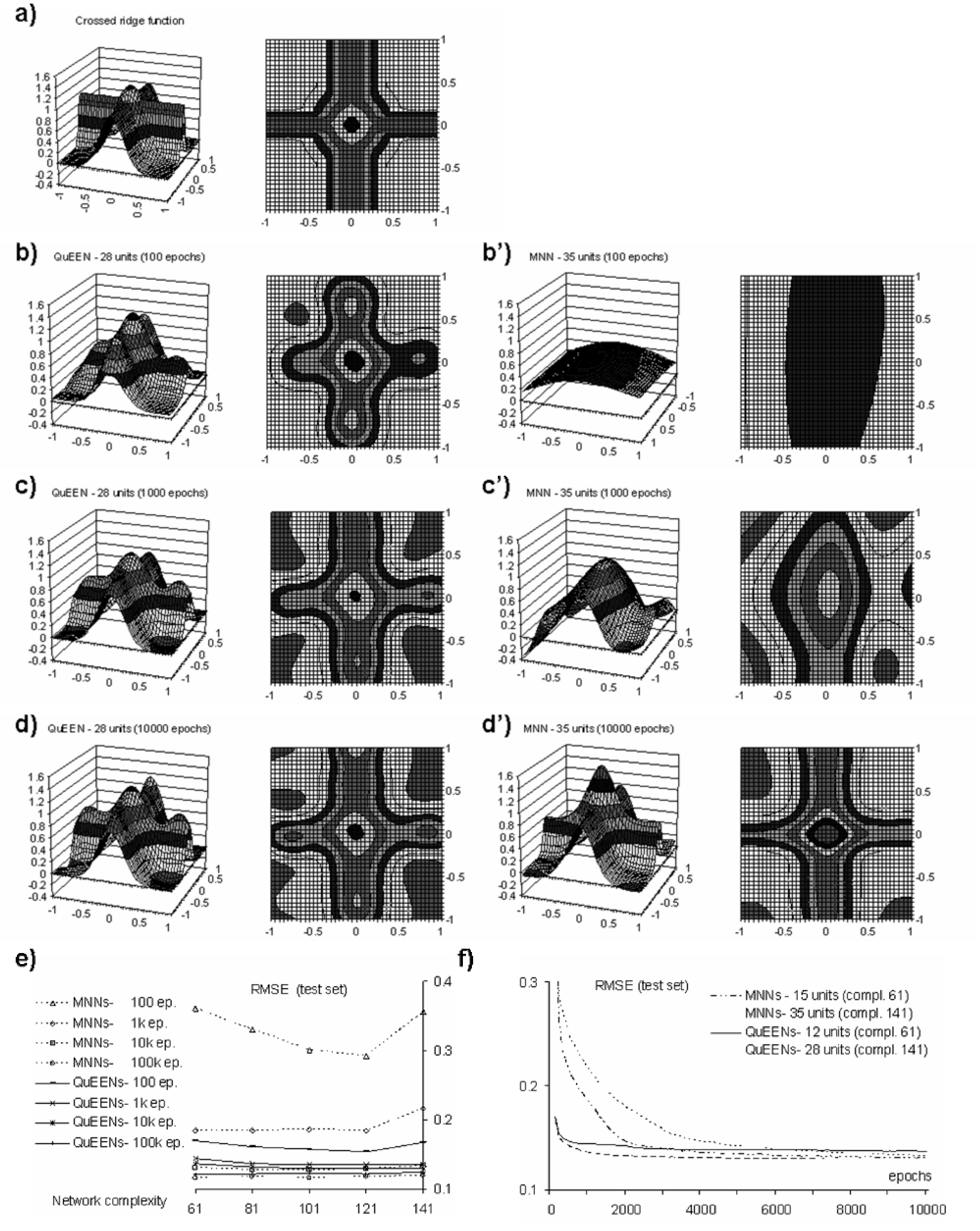


FIGURE 5.11 *Regression of the crossed ridge function by 35 units MNN and 28 units QuEEN network (compl. 141):*
a) 3D surface diagram and 2D contour diagram of the theoretical function.
b), b') Regression after 100 epochs. c), c') Regression after 1000 epochs. d), d') Regression after 10000 epochs.
e) RMSE of MNN and QuEEN networks (compl. from 61 to 141) after different epochs (test set).
f) RMSE of MNN and QuEEN networks having complexity 61 and 141 (test set).

5.5. Conclusions

We analyzed the main following drawbacks still plaguing the MNNs and EB training algorithm despite their huge knowledge and diffusion:

- the slow learning rate,
- the lack of physical meaning of the trained network,
- the negative interference, and
- the unfeasibility of parallel implementations.

We also reviewed LRFs as compactly supported kernels thus potentially able to overcome the mentioned drawbacks with particular focus on RBFNs. Unfortunately RBFNs have often been shown to be unreliable, with large size or with performance worse than MNNs because of:

- the unsupervised and sub-optimal two-phase learning that places and shapes RBFs with no exploitation of the knowledge about the desired mapping, and
- the excessive simplification given by identical and radially symmetric LRFs

Therefore, as neither the same smoothing factors, nor the radial symmetry for the LRFs are required to prove they are universal approximators, we derived the requirements an ideal LRF network should have:

1. elliptical and differently shaped LRFs,
2. joint supervised training of all the network adjustable parameters.

We thus proposed QuEEN networks based on an elliptical Gaussian LRF with different variances along each different input, and we showed that such a network may be reduced to an appropriate not fully connected two-hidden layers MNN with quadratic and exponential activations.

The standard fully supervised EB can thus be used and all the adjustable parameters of QuEEN networks – namely centers, smoothing factors and heights, having an inherent physical meaning – can be so *jointly* trained exploiting all the available information, thus meeting the preceding requirements.

Contrarily to conventional RBFN – that cannot estimate the influence of each input on output – the QuEEN networks allow determining the importance factor S of each input variable on output also derived in (Yeh, Zhang, Wu, & Huang, 2010a; Yeh, Zhang, Wu, & Huang, 2010b) for ARBF networks, so that the model resulting after the training may provide an interpretation of the modeled system.

As the weights of a QuEEN network represent QuEEN centers and asymmetric smoothing factors, the EB training of QuEEN may be regarded as a supervised clustering of the input space that takes into account the knowledge on the input-output mapping, and provides the centroids and the shape of M elliptical clusters.

As neither the numerical complexity nor the time needed to determine the LRFs centers and smoothing factors have ever been taken into account to assess the network complexity and the overall training time, we introduce a complexity given by the total number of adjustable parameters of the network, so that a comparison between the standard MNNs and the QuEEN networks may be performed in terms of regression power and learning rate.

The validity of the EB training of the QuEEN networks and their physical meaning was shown by regressing two simple functions by one- and two-QuEEN networks trained by 500 noisy samples on the [-1.0,1.0]×[-1.0,1.0] square of $\boldsymbol{R}^2$, and tested by samples extracted on a grid of the same domain. After the training, the residual regression RMSE was given by the AGN standard deviation so that the targets were perfectly learned, whereas the QuEEN networks parameters rightly reached the related target function characteristics (maximum, minimum, variances and heights). The QuEEN parameters converge faster for the input coordinates corresponding to a high steepness of the target function.

We then considered a basic bump function as a target and we showed that, to obtain the RMSE value provided by an one-QUEEN network, a MNN needs at least three hidden units, and that by fixing the same complexity, the networks have the same regression powers but the QuEENs converge faster than MNNs.

A very similar behavior was found in the regression of the more complex crown function, so outlining the advantage of using QuEEN networks as compared to MNNs, in terms of training epochs, complexity and accuracy to build a bump, and confirming (Lapedes & Farber, 1987) about LRFs.

QuEEN networks were also compared to conventional MNNs considering a well-known complex learning task given by the regression of the crossed ridge function (Schaal & Atkeson, 1998).

We still found that, when considering the same complexity, QuEEN networks showed a generally similar regression power with a considerably faster learning rate than MNNs, so confirming what found about LRF in (Lapedes & Farber, 1987; Moody & Darken, 1988).

A fully supervised learning of multivariate Gaussian kernels – referenced as hyper basis functions (HBFs) – with a generic covariance matrix and contours given by arbitrarily oriented hyper-ellipsoids in the input space was also proposed in (Poggio & Girosi, 1990a) where HBFs showed performance comparable to MNNs. QuEENs can thus be considered as separable HBFs with diagonal covariance matrix, with the axes of hyper-elliptical contour lines parallel to the axes of the input space. The proposed approach regards QuEEN networks as particular MNNs, it allows the *direct* application of *just* the standard EB and confirmed (Poggio & Girosi, 1990b) as QuEEN networks performance resulted comparable to MNNs.

Therefore the proposed QuEEN networks allow keeping the advantages of both MNNs (in terms of know-how and simplicity of EB and its variations) and LRFs (in terms of fast learning rate and physical meaning), overcoming their respective drawbacks.

Contrarily to QuEEN networks, the regression by MNNs showed some saddle-like effects (non zero values) at the corner of the target support. This behavior is an example of negative interference due to the infinite support of the sigmoidal activations.

We also found that both learning rate and regression power of QuEEN networks are improved if the initial values for the smoothing factors are chosen small with respect to the smoothness of the target function. This confirmed the results of sensitivity analysis of gradient descent that other authors performed for radially symmetrical RBFs with fixed widths (Karayiannis, 1997; Karayiannis, 1999; Karayiannis & Randolph-Gips, 2003).

LRF networks need more data to achieve a precision similar to standard MNNs and sigmoidal units perform global, rather than local, fit to the training data as found in (Moody & Darken, 1988). Even if this topic is not deepened in this work, some preliminary results allow us to strongly believe that this is also true for QuEEN networks. This is a good point to try to answer to one of the everlasting questions around which ANN is better. When data are hard to get, MNNs approach would be preferred. On the contrary, if there is plenty of data, QuEENs may be the best choice. The latter situation is e.g. commonly found in adaptive signal processing or adaptive control, where data are generally acquired at a high rate.

Anyway, the QuEEN approach may be the only choice when the regression task regards the identification of real time adaptive systems and continuous training is needed to track the system variation with strict time constraints, and fast processing is required.

Several issues have not been addressed in this paper and are left to future research and they are listed in the following.

First of all, given the results found by other authors for adaptively shaped symmetrical (Webb & Shannon, 1998) and asymmetrical LRFs (Yeh, Zhang, Wu, & Huang, 2010a; Yeh, Zhang, Wu, & Huang, 2010b), we strongly believe that QuEENs would give lower errors than the conventional RBFs, and that QuEENs would also achieve their minimum RMSE at a lower number of centers than RBFs.

As EB may be performed updating some connections and keeping fixed the others, the learning procedure for QuEEN centers, smoothing factors and heights may be separated, similarly to what suggested in (Broohmhead & Lowe, 1988) and performed in the two-phase learning of RBFN. It is thus possible to firstly train only the QuEEN centers and then, once each QuEEN is centered, keep fixed each center while only the variances are trained. If the initial QuEEN variances are chosen to be narrow enough, the first phase of training would place each QuEEN under the closer function maximum (or minimum). Further training for variances would allow each centered QuEEN to widen and correctly fit the data. Lastly, the output weights may be adapted to adjust each QuEEN height. If opportunely tuned, this learning strategy may lead to even faster learning.

With quite reasonable extra computation, it is possible to replace the standard EB by more sophisticated second order variants that generally improve convergence rates (e.g. quasi Newton algorithm) (Broohmhead & Lowe, 1988). Other variants of the EB are basically related to factors and corrections applied to the weights update equation (Rumelhart, Hinton, & Williams, 1986; Widrow & Lehr, 1990). As QuEEN networks are reduced to a MNN, the application of those well-known EB variations is straightforward.

In this work we only considered real-valued regression problems. As arbitrary decision regions can be formed as unions of convex regions, LRF networks are naturally able to separate convex regions (Moody & Darken, 1988). A classification scenario very similar to what has been applied to RBFN in (Schwenker, Kestler, & Palm, 2001) may be considered, where the number of output units L corresponds to the number of classes with class memberships encoded through a 1-of-L coding into a binary vector of $\{0, 1\}^L$. Classification is then performed by assigning the input vector to the class of the output unit with maximum activation. As the class of real-valued mappings contains classification problems as particular case, we believe that QuEENs would give even better results if classification problems formulated as a Boolean mapping task were considered.

Given both the introduction of the importance factor and the feasibility of parallel implementation of QuEEN networks, opportune pruning and growing strategies may be derived to better understand the behavior of the modeled system and reduce the computational burden. The importance of each input on the output may be evaluated even during the training in order to prune inputs with low importance. QuEENs with small heights (i.e. output weights) may be also pruned. Moreover a QuEEN may be pruned if it overlaps too much with another QuEEN (Schaal & Atkeson, 1998). As a first trivial growing strategy, further QuEENs may be added if the RMSE on the training set is too large or even during the training when the RMSE stops decreasing. QuEENs may be also added when one (or more) of the training input data do not activate any QuEEN. This growing strategy may be applied with an incremental learning scheme: the network training starts with few QuEENs and only a small subset of the training data; new QuEENs are successively added if the training error exceeds a threshold when new training data are used.

Given the fast learning time, promising applications of QuEENs regard real-time systems and online learning schemes where the training set is adaptively changed and continuous training is required as in control systems, real-time recognition and time series prediction (Zhang, Patuwo, & Hu, 1998; Gneo, Muscillo, Goffredo, Conforto, Schmid, & D'Alessio), neural based eye tracking systems (Gneo, Schmid, Conforto, & D’Alessio, 2012).

Chapter 6

Towards eye-controlled wheelchairs

ABSTRACT

The neural mapping function of the new model-independent eye-gaze tracking system proposed in Chapter 2 and Chapter 3 allows to avoid any specific model assumption and approximation either for the user's eye physiology or the system initial setup, and admits a free geometry positioning for the user and the system components. Those properties allow to investigate new fields of applications of eye-gaze tracking systems such as the control of electric-powered wheelchair.

All the methods to control electric-powered wheelchair with user's gaze require a graphical user interface (GUI) to select and confirm commands. This kind of GUI may give non natural guide and partial obstructed sight. Further gaze independent inputs are so needed for safety issues. Thanks to the flexibility of the proposed eye-gaze tracking system, a high-level scheme of a system integrating it to a brain-computer interface is conceived so to allow the user to select the desired motion command using his/her gaze, and using the user's electroencephalogram as a motion activation command, obtaining a safer obstruction-free eye- and brain guided electric-powered wheelchair[*]*.*

[*] Results described in this Chapter were published in International Journal of Bioelectromagnetism, Vol. 13, No. 1, pp. 44-45, 2011.

6.1. Introduction

A standard electric-powered wheelchair (EPW) is a wheelchair acted by an electric motor with a hand-operated joystick providing navigational controls.

Though paralyzed users who cannot use the joystick have other special devices available (touchpad, head/chin/speech control, sip-n-puff), some locked-in patients keep only very poor residual motor abilities, among which the oculomotor control is preserved for long periods (e.g. amyotrophic lateral sclerosis).

Two possible approaches allowing those patients to guide EPWs – eye-gaze tracking systems (Tuisku et al., 2008) and brain computer interfaces (BCIs) (Millán et al., 2009) – have been mainly analyzed alone. In (Zander et al., 2010) eye movements select objects and a BCI gives the mouse click on a human computer interface (HCI). Following a similar philosophy, we propose to integrate an EGTS with an EEG BCI to control EPWs.

EGTSs estimate the user's point of gaze (POG) either providing information on the oculomotor tract (e.g. in ophthalmology, neurology) or to drive input devices for HCI. While intrusive EGTSs require physical contact (e.g. contact lenses, electrodes fixed around the eye), video-based EGTS use eye images captured by cameras (Duchowsky, 2002). There are no currently available systems allowing the user to directly look where he/she wishes to go (*eyes-up* interfaces), since existing systems require the user to continuously look at a GUI to select and validate the EPW command during motion (*eyes-down* interfaces) (Tuisku et al., 2008). Thus, eye-controlled EPWs exhibit two main problems: first, as the user is always gazing at somewhere, undesired commands may be activated (the so-called Midas touch); then, the GUI hardware may obstruct visibility, and the driver always needs to stay really focused in the desired direction.

An EEG-based BCI uses the electric signal measured on the scalp to classify cortical activity and translate it into commands for a given device. Due to the noise and reduced spatial resolution, EEG-actuated devices are limited by a low information transfer rate and are generally considered too slow for controlling rapid and complex robot movements. EEG-based BCI can be, however, used as an effective binary switch for movement activation: for instance, event related de/synchronization (Pfurtscheller & da Silva, 1999 can be exploited as a method to drive this switch based on non complex motor imagery tasks.

As some authors considered eye-control still immature (Tuisku et al., 2008) and unsafe to control EPWs (independent inputs should be considered), and the BCI activation command has been shown as being more reliable (though slower) than the eye dwell time (Zander et al., 2010), we propose to use an EGTS to select the desired direction, and EEG signals to activate the motion along that direction, avoiding both the Midas touch and the need to stare at a GUI. Therefore, the user can naturally control the EPW looking at the place to be reached and activating the BCI when motion is desired, letting her/his sight free.

6.2. Materials and Methods

The high-level control signal for an EPW may be given by the desired linear and angular speeds (V_{des},Ω_{des}). A simple kinematic model of the EPW may transform it in the left and right wheel angular speeds (Ω_L,Ω_R) for a 2-wheel EPW (Barea et al., 2002).

A simplified set of 4 commands can be: *Move forward/Move backward* (V_{des} steps up/down), *Move left/right* (Ω_{des} steps up/down). A scheme of the proposed system is shown in Figure 6.1. The POG estimated by a video-based EGTS selects which among the 4 possible commands the user desires, so the (V_{des},Ω_{des}) couple can be calculated. A visual lighting scheme can provide the user with a feedback about the selected command (e.g. 4 LEDs, placed as cardinal points near the camera and/or within the user's peripheral sight). Depending on the received feedback, the user validates and activates (or ignores) the command via the EEG-BCI. The visual feedback scheme guides also the EGTS calibration asking the user to look at 4 known directions corresponding to the possible commands.

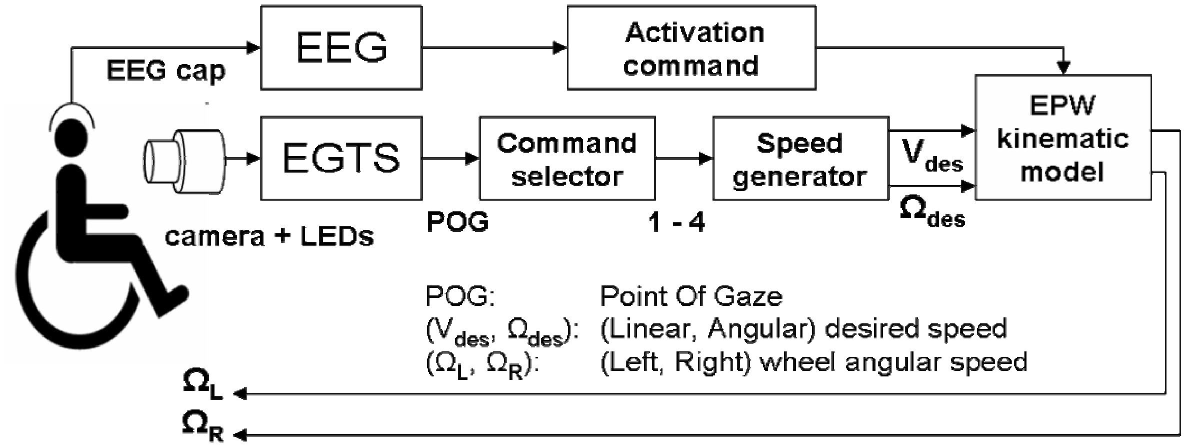


FIGURE 6.1 *High-level scheme of the proposed guidance system for electric-powered wheelchairs*

Given its large flexibility, the geometry-free EGTS recently presented in Chapter 2 (Gneo et al., 2010) and Chapter 3 (Gneo et al., 2012) can be easily assembled on an EPW. Since asynchronous BCIs are showed able to continuously control mobile robots in a house-like environment (Millán et al., 2004), the asynchronous EEG-based binary switch based on the method proposed in (Townsend et al., 2004) can be used as the activation command generator. This unit will be activated only when the user gets a positive feedback, i.e. when the command selected by the EGTS matches the desired command. In this contribution, a preliminary EGTS testing phase on two healthy subjects was performed.

6.3. Results and Discussion

With reference to the free geometry model-independent EGTS proposed in Chapter 3 (Gneo et al., 2012), we considered a system configuration having the illumination system and the camera displaced respectively 30° to the left and right of the user's sagittal plane. That system configuration allows a convenient integration of the EGTS with the EPW so that the user sight is not obstructed.

Tests performed on such a configuration of the EGTS, showed 0.44° and 0.41° for the horizontal and vertical POG estimation accuracies, respectively.

6.4. Conclusions

A high-level scheme is proposed to integrate an EGTS and a EEG-BCI to control an EPW, overcoming the problems exhibited by eye-controlled EPWs (Midas touch, eyes-down interface) and BCI (slow command rate), and augmenting its safety (BCI-activation is a conscious, explicit command in contrast to the implicit commands of dwell time solutions).

A convenient configuration of the free geometry model-independent EGTS proposed in (Gneo et al., 2012) is considered so that the user visibility is not obstructed.

Preliminary tests on this EGTS configuration showed a horizontal and vertical POG estimation accuracies (0.44° and 0.41°, respectively) proving that the proposed EGTS can be conveniently integrated with an EEG-based binary switch derived from (Townsend et al., 2004).

This will be object of future research.

General Conclusions

Contrarily to the model based eye gaze tracking systems, model-independent methods estimate the mapping function by means of regression techniques with no need of any specific model assumption and approximation either for the user's eye physiology or the system initial setup.

In this context, the results presented in this dissertation confirm the suitability of artificial neural networks for the estimation of the mapping function.

According to the consolidated pupil center corneal reflection technique, the coordinates of the pupil and outer corneal reflections (i.e. glints) of the user's eye images are mapped onto the coordinates of her/his gaze on the observed surface.

A new model-independent eye gaze tracking system based on this technique has also been conceived and built. The neural mapping function gives the system the property to admit a free geometry positioning for the user and the system components, whereas the triangular pattern given by the particular configuration of the proposed illuminating system increases the robustness of the detection of pupils and corneal reflections.

New architectures and learning scheme for artificial neural networks have also been proposed to further improve the system robustness and the mapping function learning rate during the calibration.

Thus following the rationale of reducing the effects of the main drawbacks still plaguing each studied or sold eye tracking system, avoiding the assumption (and the inherent approximation) of any explicit model, simplifying the system architecture, and hopefully increasing its accuracy and robustness, this thesis specifically dealt with the following aspects:

1. the proposal of a new model-independent (neural based) eye-gaze tracking system equipped with a innovative simplified illuminating system,
2. the real-time time series prediction based on neural networks, in order to overcome the problems due to failures in eye features detection,
3. the introduction of artificial neural networks of new localized receptive fields given by elliptical neuron in order to give physical meaning to the model built during the calibration with similar regression power and faster learning rate than conventional neural networks,
4. a high-level scheme of a system integrating the proposed eye-gaze tracking system and a conventional brain-computer interface into an electric-powered wheelchair to allow the user to select the desired motion command using his/her gaze, using the user's electroencephalogram as a motion activation command.

With respect to the first aspect, the prototype of an eye-gaze tracking system equipped with an innovative illuminating system and estimating the mapping function by means of artificial neural networks was built. Three sources of lights generate a triangular pattern of three glints on the user's eye and avoid the synchronization with the image capturing system, whereas the use of artificial neural networks allows to directly evaluate the mapping function and avoids the assumption of any explicit model, so giving a geometry-free system.

The feasibility of the proposed system was proven in Chapter 2, where the successful tests performed during several sessions of real operation are reported.

The robustness of the proposed system was also proven in detail in Chapter 3 by assessing its accuracy when tested on real data coming from: i) different users; ii) different geometric settings of the camera and the light sources; iii) different protocols based on the observation of points on a calibration grid and halfway points of a test grid. The achieved accuracy is not greater than 0.49°, 0.41°, and 0.62° for respectively the horizontal, vertical and radial error of the point of gaze. Then, the actual system performs better than eye-gaze tracking systems designed for human computer interaction which, even if equipped with superior hardware, show accuracy values in the range 0.6°-1°.

With respect to the second aspect, in order to overcome the problems due to failures in eye features detection in eye-gaze tracking systems, a real-time time series prediction based on the neural networks used to regress the mapping function was proposed in Chapter 4. That prediction scheme was successfully validated applying it to the gesture recognition considering the time series given by the output of two accelerometers placed on the upper arm and on the forearm respectively. The prediction errors are used both to train the neural networks and estimate a measure of the unlikelihood of the specific gesture occurrence. According to the model-independent approach, neither a priori assumptions nor signal pre-processing is performed. On the four significant gestures considered, the proposed method showed a correct recognition rate higher than 83%. That encourages the future integration of the described prediction scheme into the mapping function of the previously proposed eye-gaze tracking system.

The infinite support of sigmoidal activations of conventional multilayer neural networks causes slow learning rate, lack of physical meaning, negative interference. This may prevent the useful application of artificial neural networks on eye tracking giving, in particular, slow calibrations. Localized receptive field networks have promised similar regression power and faster learning than multilayer neural networks, and physically meaningful modeling but, unfortunately, they have often large size and/or performance worse than multilayer neural networks due to unsupervised placing and shaping of identical and radially symmetric kernels.

With respect to the third aspect, new elliptical localized receptive fields giving similar regression power and faster learning rate were introduced in Chapter 5. As networks of the proposed localized receptive fields, called quadratic exponential elliptical neurons (QuEENs), can be reduced to opportune multilayer neural networks, the standard error backpropagation allows each neuron to be self placed and shaped by a supervised training. According to simulations, QuEEN networks showed comparable regression power and faster learning than multilayer neural networks.

Furthermore, with reference to the last issue, as the neural mapping function of the proposed eye-gaze tracking system allows to avoid any specific model assumption and approximation either for the user's eye physiology or the system initial setup, and admits a free geometry positioning for the user and the system components, a promising application of the proposed system to control an electric-powered wheelchair with user's gaze was analyzed in Chapter 6. All similar systems require a graphical user interface to select and confirm commands. This kind of interface may give non natural guide and partial obstructed sight. Further gaze independent inputs are so needed for safety issues. Thanks to the flexibility of the proposed eye-gaze tracking system, a high-level scheme of a system integrating it to a brain-computer interface was conceived so to allow the user to select the desired motion command using his/her gaze, and using the user's electroencephalogram as a motion activation command, obtaining a safer obstruction-free eye- and brain guided electric-powered wheelchair.

Some important drawbacks still plague studied or sold eye-gaze tracking systems and much needs to be done: the results obtained in this thesis could hopefully offer several useful issues and hints to overcome those problems related to the scarcity of the robustness and accuracy mainly due to the approximation inherent in the assumptions related to each system.

Towards that direction, given the results reported in this thesis, future work should be planned to apply all the learning scheme and architecture of artificial neural networks to model-independent eye-gaze tracking systems based on neural networks.

List of Acronyms

ALS	Amyotrophic lateral sclerosis
ANN	Artificial Neural Network
AR	Auto Regressive
BCI	Brain Computer Interface
CCD	Charge-Coupled Device
CRPR	Correct Recognition Percentage Rate
cSMAPE	Corrected Symmetric Mean Absolute Percentage Error
EB	Error Backpropagation
EEG	Electroencephalogram
EGTS	Eye-Gaze Tracking System
EPW	Electric-Powered Wheelchair
FNN	Feedforward Neural Networks
GRNN	Generalized Regression Neural Network
GUI	Graphical User Interface
HBF	Hyper Basis Function
HCI	Human-Computer Interaction
HT	Hough Transform
ILED	Infrared Light Emission Diode
IPP	Integrated Performance Primitive
IR	Infrared light
LED	Light Emitting Diode
LRF	Localized Receptive Field
LRFN	LRF network
MAD	Mean Absolute Deviation

MAPE	Mean Absolute Percentage Error
MNN	Multilayer Neural Network
MSE	Mean Squared Error
PCCR	Pupil Center Corneal Reflection
POG	Point Of Gaze
POR	Point Of Regard
QuEEN	Quadratic Exponential Elliptical Neuron
QuEENN	Quadratic Exponential Elliptical Neuron Network
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
RSD	Relative Standard Deviation
RTNP	Real-Time Neural Predictor
SMAPE	symmetric mean absolute percentage error
VOG	Video-oculography
WMFT	Wolf Motor Function Test

References

Baluja, S. & Pomerleau, D. (1994). Non-intrusive gaze tracking using artificial neural networks. Technical Report, CMU-CS-94-102, Carnegie Mellon University.

Barea, R., Boquete, L., & Mazo, M. (2002). System for Assisted Mobility Using Eye Movements Based on Electrooculography. In *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(4): 209-218.

Benoudjit, N. & Verleysen, M. (2003). On the kernel widths in radial-basis function networks. Neural Processing Letters, 18(2), 139–154.

Bishop, C. M. (1994). Neural networks and their applications. *Review of Scientific Instruments*, 65(6), 1803-1832.

Broomhead, D. S. & Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Syst.*, 2, 321-355.

Brown, M. (1996), Layered perceptron networks and the error back propagation algorithm. Architecture, 1-38.

Chen, S., Cowan, C. F. N., & Grant, P.M. (1991). Orthogonal least squares learning algorithm for radial basis function networks. IEEE Trans. Neural Networks, 2(2), 302–309.

Chiu, C. C., Cook, D. F., Pignatiello, J.J., & Whittaker, A. D. (1997). Design of a radial basis function neural network with a radius-modification algorithm using response surface methodology. Journal of Intelligent Manufacturing, 8(2), 117-124.

Crone S. (2005). Stepwise selection of artificial neural network models for time series prediction. *J Intell. Syst.*, 14(2-3):99-122.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.*, 2(4), 303-314.

Droege D., Geier T., & Paulus, D. (2007). Improved low cost gaze tracker. In *Proc. Conf. COGAIN 2007*, 37-40.

Duchowski, A. T. (2002). A breadth-first survey of eye tracking applications. In *Behavior Research Methods, Instruments, & Computers*, 34(4), 455-470.

Duchowski, A. T. (2007). Eye Tracking Methodology: Theory & Practice, Springer-Verlag, London, UK, 2nd edition.

Duda, O. R. & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1), 11–15.

Ebisawa, Y. (1998). Improved video-based eye-gaze detection method. *IEEE Trans. Instrumentation Meas.*, 47(4), 948-955.

Giansanti, D., Macellari, V., Maccioni, G., et al. (2003). Is it Feasible to Reconstruct Body Segment 3-D Position and Orientation Using Accelerometric Data. *IEEE Trans. on Biomed. Eng.*, 50(4).

Gneo, M., Carbone, D., Schmid, M., Conforto, S., Palma, C., & D'Alessio, T. (2010). Feasibility of a new Geometry-Free Eye Gaze Tracking using a new Triangular Pattern of Infrared Light and Neural Mapping. In proc. *National Congress of Bioengineering*, Ed. Pàtron, pp. 657-658.

Gneo, M., Muscillo, R., Goffredo, M., Conforto, S., Schmid, M., & D'Alessio, T. (2009). Real-time adaptive neural predictors for upper limb gestures blind recognition. In *Proc. IFMBE Proceedings World Congress on Medical Physics and Biomedical Engineering*, 25(9), 536-539.

Gneo, M., Schmid, M., Conforto, S., & D'Alessio T. (2012).A Model Independent and Free Geometry Neural Eye-Gaze Tracking System," submitted to *IEEE Transactions on Systems, Man, and Cybernetics Part B*

Guestrin, E. D. & Eizenman, M. (2006). General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Trans. Biomedical Eng.*, 53(6), 1124-1133.

Guillén, A., Rojas, I., González, J., Pomares, H., Herrera, L. J., Valenzuela, O., & Rojas, F. (2007). Output value-based initialization for radial basis function neural networks. *Neural Processing Letters*, 25(3), 209-225.

Hansen, D. W. & Ji, Q. (2010). In the eye of the beholder: a survey of models for eyes and gaze. *IEEE Trans. Pattern Anal. Machine Intell.*, 32(3), 478-500.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.

Hutchinson, E. H., White, K. P. Jr., Martin, W. N., Reichert, K. C., & Frey, L. A. (1989). Human-computer interaction using eye-gaze input. *IEEE Trans. Systems, Man, Cybernetics*, 19(6), 1527–1534.

Karayiannis, N. B. & Randolph-Gips, M. M. (2003). On the Construction and Training of Reformulated Radial Basis Function Neural Networks. *IEEE Trans. Neural Networks*, 14 (4), 835-846.

Karayiannis, N. B. (1997). Gradient descent learning of radial basis neural networks. In *Proc. 1997 IEEE Int. Conf. Neural Networks*, 3, 1815–1820.

Karayiannis, N. B. (1999). Reformulated radial basis neural networks trained by gradient descent. *IEEE Trans. Neural Networks*, 10, 657–671.

Lapedes, A.S. & Farber, R.M. (1987). How Neural Nets Work. *Neural Information Processing Systems - NIPS Conf.*, 442-456.

LC Technologies Inc., Eyegaze Systems. [Online]. Available:
http://www.eyegaze.com/content/instrument-specifications.

Martinez-Conde, S., Macknik, S. L., & Hubel, D. H. (2004). The role of fixational eye movements in visual perception. *Nature Reviews Neuroscience*, 5, 229–240.

Mathie, M. J., Coster, A. C., Lovell, N. H., et al. (2004). Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiol. Meas.*, 25, R1-20.

Millán, J. del R., Galán, F., Vanhooydonck, D., Lew, E., Philips, J., & Nuttin, M. (2009). Asynchronous Non-Invasive Brain-Actuated Control of an Intelligent Wheelchair. *31st Annual Intern. Conf. IEEE Engineering in Medicine and Biology Society*, 3361-3364.

Millán, J. del R., Renkens, F., Mouriño, J., & Gerstner, W. (2004). Brain-Actuated Interaction. *Artificial Intelligence*, 159: 241–259.

Moody, J., & Darken, C. (1988). Learning with localized receptive fields. Proceedings of the *1988 Connectionist Models Summer School*, Hinton, Sejnowski, and Touretzsky, eds. Morgan Kaufmann, pp. 133-143.

Moody, J., & Darken, C. (1989). Fast-learning in networks of locally-tuned processing units. *Neural Computation*, 1(2), 281-294.

Morimoto, C. H. & Mimica, M. R. M. (2005). Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst.*, 98(1), 4-24.

Morimoto, C. H., Koons, D., Amir, A., & Flickner, M. (2000). Pupil detection and tracking using multiple lighting sources. *Image Vision Computing*, 18(4), 331-335.

Muscillo, R., Conforto, S., Schmid, M., Caselli, P., & D'Alessio, T. (2007). Resolving ADL Activities Variability through Derivative Dynamic Time Warping applied on Accelerometer Data. *IEEE EMBS Proc. 29th Intern. Conf. 2007*, 4930-33.

Neural Forecasting Competition [Online].
Available: http://www.neural-forecasting-competition.com

Park, J., & Sandberg, J. W. (1991). Universal approximation using radial basis functions network. *Neural Computation*, 3, 246–257.

Park, J., & Sandberg, J. W. (1993). Approximation and radial-basis-function networks. *Neural Computation*, 5, 305–316.

Pentland, A. (2005). Healthwear: medical technology becomes wearable. *Stud. Health Technol. Inform*. 118:55-65.

Pfurtscheller, G., & Lopes da Silva, F. H. (1999). Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clin. Neurophysiol*. 110(11):1842-57.

Piratla, N. M., & Jayasumana ,A. P. (2002). A neural network based real-time gaze tracker. *J. Network Computer Appl.* 25(3), 179–196.

Poggio, T., & Girosi, F. (1990a). Networks for approximation and learning. *In Proceedings of the IEEE*, 78(9), 1481-1497.

Poggio, T., & Girosi, F. (1990b). Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247(4945), 978-982.

Powell, M. J. D. (1987). Radial basis functions for multivariate interpolation: a review. *In Mason*, *J. C., & Cox. M. G. (Eds.), Algorithms for Approximation*, Oxford: Clarendon Press 143-167.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In: D.E. Rumelhart and J.L. McClelland, Editors, *Parallel Distributed Processing*, MIT Press, Cambridge, MA, 1(8), 318–362, 1986

Scarselli, F., & Tsoi, A. C. (1998). Universal approximation using feed-forward neural networks: a survey of some existing methods, and some new results. *Neural Networks*, 11(1), 15-37.

Schaal, S., & Atkeson, C. G. (1998). Constructive incremental learning from only local information. *Neural Computation*, 10(8), 2047-2084.

Schasfoort, F. C., Bussmann, J. B. J., & Stam, H. J. (2002). Ambulatory measurement of upper limb usage and mobility-related activities during normal daily life with an Upper Limb-Activity Monitor: a feasibility study. *Med. Biol. Eng. Comput.*, 40(2):173-82.

Schwenker, F., Kestler, H. A., & Palm, G. (2001). Three learning phases for radial-basis-function networks," Neural Networks, vol. 14, no. 4-5, pp. 439-458.

Tobii Technology, Danderyd, Sweden. [Online].
Available: http://www.tobii.se/

Torricelli, D., Conforto, S., Schmid, M., & D'Alessio, T. (2008). A Neural-Based Remote Eye Gaze Tracker under Natural Head Motion. *Computer Methods and Programs in Biomedicine*, 92(1), 66-78.

Townsend, G., Grainmann, B., & Pfürtscheller, G. (2004). Continuous EEG classification during motor imagery-simulation of an asynchronous BCI. In *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12(2): 258–265.

Tuisku, O., Bates, R., Stepankova, O., Fejtova, M., Novak, P. Istance, H., Corno, F., & Majaranta, P. (2008). D2.6 A survey of existing 'de-facto' standards and systems of gaze based mobility control. In *Communication by Gaze Interaction (COGAIN)*.

Veltink, P. H., Bussmann, H. B., de Vries, W., Martens, WimL. J., & Van Lummel, R.C. (1996). Detection of static and dynamic activities using uniaxial accelerometers. *IEEE Trans. Rehabil. Eng.*, 4, 375-85.

Villanueva, A., & Cabeza, R. (2008). A novel gaze estimation system with one calibration point. *IEEE Trans. Systems, Man, and Cyber.* — Part B, 38(4), 1123-1138.

Webb, A. R., & Shannon, S. (1998). Shape-Adaptive Radial Basis Functions. *IEEE Trans. Neural Networks,* 9(6), 1155–1166.

Widrow B., & Lehr, M. A. (1990). 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. In *Proc. IEEE*, 78(9), 415-1442.

Yeh, I., Zhang, X., Wu, C. & Huang, K. (2010b). Adaptive radial basis function networks with kernel shape parameters. *Neural Computation & Applic.*, 1-12.

Yeh, I., Zhang, X., Wu, C., & Huang, K. (2010a). Radial basis function networks with adjustable kernel shape parameters. *In Proc. ICMLC*, 3, 1482-1485.

Yoo, D. H., & Chung, M. J. (2005). A novel non-intrusive eye gaze estimation using cross-ratio under large head motion. *Computer Vision and Image Understanding*, 98(1), 25–51.

Zander, T. O., Gaertner, M., Kothe, C., & Vilimek, R. (2010). Combining Eye Gaze Input with a Brain-Computer Interface for Touchless Human-Computer Interaction," [Online]. Available: http://futureofeeg.wdfiles.com/local--files/start/Zander4_gaze.pdf).

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *J Forecasting*, 14:35–62.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (2001). A simulation study of artificial neural networks for nonlinear time-series forecasting. *Comput. Oper. Res.*, 28:381-396.

Zhu Z., & Ji, Q. (2007). Novel Eye Gaze Tracking Techniques Under Natural Head Movement. *IEEE Transactions on Biomedical Engineering*, 54(2), 2246–2260.

Zhu, Z., & Ji, Q. (2004). Eye and gaze tracking for interactive graphic display. *Machine Vision Applications*, 15, 139–148.