DUBLIN CITY UNIVERSITY

Ollscoil Chathair Bhaile Átha Cliath

DUBLIN CITY UNIVERSITY

SCHOOL OF ELECTRONIC ENGINEERING

M.ENG. IN ELECTRONIC ENGINEERING

Supervised by Dr. John Ringwood

An Online Learning Environment to Teach Artificial Neural Networks

Gareth Galvin B.Eng.

March 1999

Acknowledgements

I would like to thank the following people:

Dr. John Ringwood for his guidance and support throughout this project

Guy-Michel Cloarec for his help and support.

The MEng class who took the ANN module for the interest and feedback throughout the project.

Dave, Sherdy, Denise, Paul and Muggy, to name but a few, who got me through this in one piece.

And last but not least, my family and friends for their support and encouragement throughout my life.

Declaration

I hereby certify that this material, which I now submit for assessment on the programmme of study leading to the award of Masters in Electronic Engineering is entirely my own work and has not been taken from the work of others save to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Gareth Galvin Date: 5/3/99

Contents

List of Figures

List of Tables

Acronyms/Abbreviations

- ANN - Artificial Neural Networks
- BTC - Benefit-to-Cost
- CAL - Computer Aided Learning
- CGI - Common Gateway Interface
- CPD - Continuous Professional Development
- DCU - Dublin City University
- FTP - File Transfer Protocol
- HTML - Hypertext Mark-up Language
- MathML - Mathematical Mark-up Language
- MLP - Multi-Layer Perceptron
- MSE - Mean Square Error
- PC - Personal Computer
- PCA - Principal Components Analysis
- RBF - Radial Basis Function
- SGML - Standard Generalised Mark-up Language
- SOFM - Self-Organising Feature Map
- SSI - Server Side Include
- SVD - Singular Value Decomposition
- WWW - World Wide Web
- W3C - World Wide Web Consortium

Abstract

An Online Learning Environment to Teach Artificial Neural Networks

Submitted by: G. Galvin

This thesis presents a study to evaluate the benefits of the Internet as an environment for Computer Aided Learning (CAL). It comprises of research into, current and previous approaches to CAL, and presents an implementation of the authors own approach, with the MEng (Masters in Electronic Engineering) course of Artificial Neural Networks (ANN's) as a template upon which to develop courseware.

CAL strives to stimulate the user interactively so as to facilitate an optimum knowledge transfer between teacher and student. This project will show that the Internet offers such features as hypertext, graphics, sound and video, which when incorporated correctly can engage the student and provide a more intuitive learning experience.

Artificial Neural Networks were chosen as a template subject because of its iterative nature which lends itself well to graphical analysis. This, combined with its computationally intensive algorithms, lends itself well to a CAL environment.

An evaluation of the suitability of the Internet as a CAL environment is carried out based on the courseware developed and the feedback of those who actively used the package as part of their study program. Based on these results the author intends to show the benefits of such a medium within the education system.

Chapter 1

Introduction

1.1 Introduction

This project examines the potential of the Internet with regard to its suitability as a Computer Aided Learning (CAL) environment. It aims to display the benefits of multimedia and interactivity as aids to the learning process and in particular the suitability of Engineering and Science based subjects to this mode of delivery. The MEng (Masters degree in Electronic Systems) program at Dublin City University (DCU), in particular the module of Artificial Neural Networks (ANNs) is used as a template upon which to develop courseware.

This project presents research into the area of CAL, in particular Internet based CAL or Web Based Teaching (WBT). Having established the Internet as a suitable medium upon which to develop a CAL environment, a practical implementation is presented for the development of CAL courseware, in particular ANN courseware. This process involves the development of the course material, the structuring of this material within an ANN web site, the incorporation of interactive examples and finally an evaluation of the courseware produced.

The online courseware developed was to be used as supplemental material for those taking the course through traditional, on-campus methods but also as primary material for those undertaking the course via the Department of Electronic Engineering's distance education program, Remote Access to Continuing Engineering Education (RACeE). As a result, a stable, working environment had to be in place for those undertaking the ANN module.

1.2 Project Background

This project is a natural progression from the Masters project "Internet-based Computer-Aided Learning for Artificial Neural Networks" [1]. This project investigated the use of the Internet as a medium for CAL and focused on the use of Java and Java Demonstrations as a means of providing interactivity. Java was used as the computational engine around which to develop these examples. Demonstrations explaining key areas of the course were developed, allowing the students to change parameters and view, in graphical form, the results of their actions.

The use of MATLAB™ as a computational engine for online demonstrations and examples was examined in the final year project "Computer Aided Learning Tool" [2]. Through the use of VCLab software [3] and Java, a communication link was set-up between the Web Browser and the computational engine of MATLAB™. The VCLab software essentially consisted of a MATLAB™ Plugin, which allowed values to be passed to and from MATLAB™ via the browser. This software, while successful in its approach was and still is in its developmental stage and so proved too unstable to use for this project.

Both projects benefited from the suitability of the Internet as a medium for an interactive CAL environment. This project proposes to, having researched the issues involved, use this Internet platform to develop a fully functional online learning environment, incorporating the interactive features contained above in addition to other tools now widely available over the World Wide Web (WWW), to provide the student with a self-contained course from which to study and explore the material being presented.

1.3 Computer Aided Learning

1.3.1 CAL and the Internet

The Internet has become a very significant electronic resource for professionals in a broad spectrum of disciplines. Since the emergence of the Mosaic Web browser in 1993 [4] interest in, and use of the World Wide Web, a subset of the Internet, has grown astronomically (increasing by as much as 10% a month [5]). This has lead to considerable research into the potential of this resource within an educational context.

The ability to include various media types such as graphics, audio and, increasingly, video offers great potential for the development of interactive courseware. The web also supports high level programming languages such as Java which can be used to develop interactive examples which involve the student directly in the topics being presented. It is accessible to anyone with a computer, modem and Internet access.

1.3.2 Artificial Neural Networks

The motivation behind Artificial Neural Networks stems from the recognition that the brain computes in a completely different way to that of conventional digital computers [6]. The modern era of neural networks is said to have begun with the pioneering work of McColloch and Pitts(1943) [7]. They devised the McColloch-Pitts neuron (a single independent computing element), which was later to be modified by Frank Rosenblatt in 1958 to form the Perceptron.

An artificial neural network (ANN) is an information processing system that has certain performance characteristics in common with biological neural networks [8]. They are networks that consist of highly interconnected processing units called neurons. These neurons, when connected in various network topologies, can be trained to solve various classification or function approximation problems.

Neural networks are adaptive systems. They use an iterative process to determine a solution. This solution is often uncertain and so cannot be solved by analytical means. For such systems, simulations and graphical analysis can greatly improve their understanding. Neural networks also require considerable computing power for training. Such characteristics lend themselves well to CAL.

The Neural Networks module of the M.Eng program at DCU is entering its second year. With the course being offered remotely, under the RACeE program, the Internet will be shown to provide an ideal environment both in terms of suitability and access upon which to present this module. This project is therefore a practical and relevant endeavor both for ANN instruction and CAL.

1.4 Thesis Structure

This thesis presents research into online CAL, a practical implementation of a CAL package to teach Artificial Neural Networks and, finally, an evaluation of the said package. It consists of nine Chapters, the motivation for which is outlined below.

The background to CAL, and the CAL environment itself, are important considerations when developing any CAL package. They influence both the structure and content, and when understood and applied properly, can enhance the overall learning experience of the user. Chapter 2 outlines some of these issues, and also examines some of the different modes of education delivery and their suitability to a CAL environment.

CAL, and in particular Internet CAL, is a growing field of research and there are many approaches as to how it can be implemented. Bearing in mind the findings of Chapter 2, some of these approaches are introduced, along with some commercially available software for authoring courseware. This software is evaluated with regard to the current needs of this project, and having done so, the author presents a proposed structure for the online neural networks course, to be named ANNet.

The layout and features of an online course are determined by the subject matter of the course. Thus, the subject of artificial neural networks is described with particular focus on its suitability for computer aided delivery. Based on the characteristics of the ANN course, a course layout is developed and and several components are described that help to convey the concepts and requirements of the course.

Having determined a layout and suitable components for the ANNet course, Chapter 5 explains their development and the properties of the Internet that contribute to this process. The software tools are described and their motivation outlined.

With a working course that embodies the CAL paradigm now developed, an evaluation of these concepts can be performed. Based on student feedback and usage statistics generated during the life of the course, the success of the approach taken by the author can be determined. In the light of these finding a cost benefit study is performed to relate the cost in terms of effort, time and expertise required to develop the package and its components, against their perceived benefits as seen by the students.

There are also practical issues associated with such a CAL package, notably that of maintenance. Chapter 8 examines issues such as the ease and cost of modifying and updating material against the commercially available courseware package.

Finally, conclusions can be drawn on the value and success of such an online CAL package both in terms of ANN instruction and also in terms of Internet CAL in general. Some proposals and recommendations for the future development are suggested.

Chapter 2

Motivation for CAL

2.1 Introduction

It is apparent that the development of modern advanced technologies combining computation and communication will offer exciting opportunities to educators [9]. Engineering disciplines, whose theory by nature can be computational intensive and whose understanding can be enhanced through graphical analysis, stand to benefit considerably from such advances. Traditional modes of information delivery such as textbook, blackboard and handouts haven't changed much over the last three decades [10]. Interactive multimedia approaches to teaching can provide a valuable step forward in this field of education.

2.2 Computer Aided Learning

2.2.1 Introduction to Computer Aided Learning

With the growing number of computers in society, research into the role they can play in the education system has become widespread. CAL offers added benefit over the traditional classroom approach by providing the student with an environment which they can have control over. It gives the student the ability to work their way through the material at a pace which is suitable to their learning needs. CAL also offers the opportunity to enrich the course material with graphics, animations and simulations, which the traditional, "chalk and talk", classroom cannot, although a increasing number of classrooms are equipped with overhead projectors which allow the educator to display previously generated graphs or diagrams. A common approach to incorporating these features is the use of Multimedia. Multimedia comes from the Latin multi - meaning many and media - meaning ways of presenting information. In relation to CAL, it is defined as a computer tool that includes a combination of text, graphics, sound and animation sequences packaged together. Providing such a combination as well as a degree of interactivity can improve the attention of the student significantly, thus promoting learning interest and improved retention of knowledge [11]. The result is an interactive audio/visual presentation of information and knowledge.

Engineering is a graphics oriented discipline in which drawings, sketches and graphs have always played an important role but require enormous manpower to produce [12]. Computers offer the computational powers to overcome this and when used within a CAL environment have the potential to greatly increase student understanding by allowing them to explore complex ideas in a graphical and hands-on way.

2.2.2 Interactivity

Studies have shown that a person retains 25% of what they hear, 45% of what they hear and see and 70% of what they do [13]. This *"70% of what they do"* refers to direct interaction and participation in what is being presented to them. Thus from the point of view of developing a CAL environment, the provision for interaction between both user and package, and indeed user and other users, is vital.

By allowing the user to have control over the flow of information, they are in fact interacting with the material and participating directly in its exploration. This enables them to extract knowledge in a way that suits their learning behaviour.

Hypertext is a concept referring to the presentation of stored information in a non-linear way as a computer-based document. Links are made between associated chunks of information, and database type products result in which the user can explore and browse for particular items [14]. Tolhurst [15] shows in Figure 2.1, in graphical form, the relationship between the *nodes of information* (capital letters in ovals) and the *physical links* between them (lower case letters).

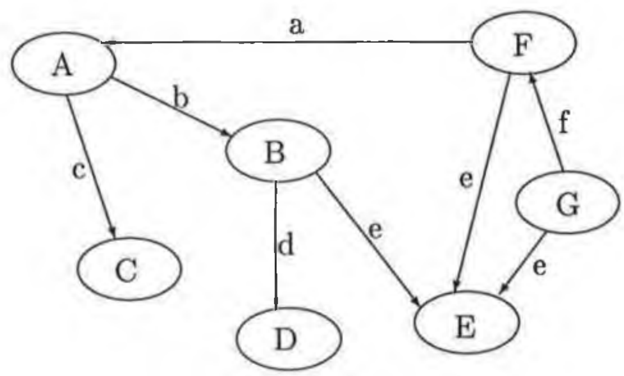


Figure 2.1: Relational diagram of nodes of information and the physical links between them

There are many ways of reaching the same piece of information. Indeed, not all the information need be used by all the students. Those who have a good background in the material might not need the same depth of explanation as those who are new to the area.

This information is not constrained to text only. Multimedia can also be incorporated into the *hyperlinked* documents. This is often referred to as hypermedia. Hypermedia gives us a better idea how to use the multimedia resources mentioned in Section 2.2.1. Whereas, "*multimedia*" indicates only that the variety in modalities of information is big, the term "*hypermedia*" implicitly advocates how to access information elements and how to crisscross in information space [16]. Within such hypermedia rich environments, extra navigational tools are required to enable the user to determine their location within the material. A poorly laid out network of material can lead to frustration on the part of the user. They may be left unsure as to whether they have covered all the necessary material [17].

Through the use of hypertext and hypermedia students can interact in a beneficial way with the course material. Students of all abilities can benefit from its self-paced approach and all levels of understanding can be catered for.

2.2.2.1 Computer Testing

Continuous testing is a useful and productive gauge of what a student has learned. Within the context of a CAL environment, such a computer based testing system can be used in two ways. The first approach is that suggested by *Gagne et al* [18]. This approach presents the information to be learned and then proceeds to ask questions on the material. Based on the answers given, the user is directed to either study further (incorrect answers) or proceed to the next block of information (correct answers) in a sequential manner, thus directing the user along a set path of study. This is illustrated in Figure 2.2.

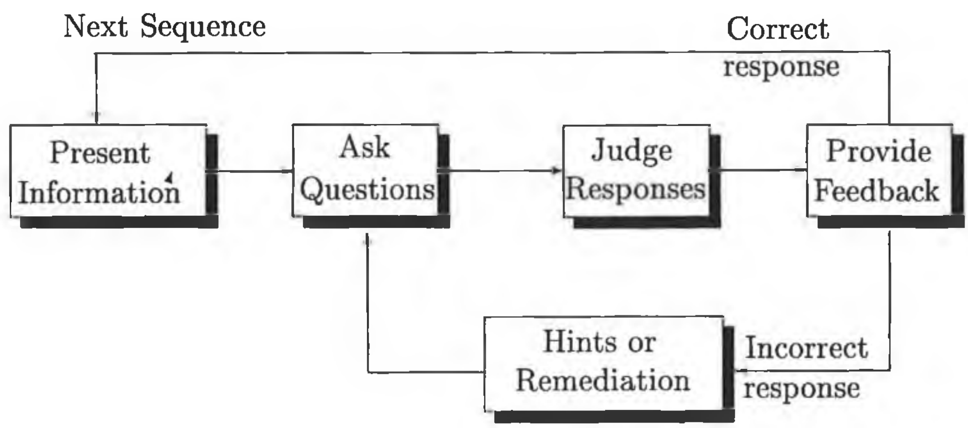


Figure 2.2: Procedure for tutorial program. Reproduced from Gagne et al [18]

The second approach, while similar in format, does not direct the user after completing the test. They may be given details of the performance or hints/remediation for incorrect answers, but they are not forced to follow a set path, thereby maintaining the non-linear, hypertext approach to exploring the material. The test merely serves as an indicator for the student as to their level of understanding of the information presented.

2.2.3 The CAL Environment

When considering the development of a CAL package, an important factor is the type of environment upon which to host the package. As discussed in Section 2.2.2, interactivity and multimedia capabilities are important to the learning process. The ease of incorporation and development of these features must be considered.

The ability to update and add new material with minimal disruption to both lecturer and student is vital to ensure the smooth running of the course. Accessibility is another important consideration. It is essential that students be able to access the material regardless of their location or the type of computer hardware they have.

Communication between student and lecturer is necessary so as the student does not feel isolated. They should be able to pose problems or queries to the lecturer and receive a quick and thorough response (diagrams etc where necessary). This *rapid feedback* is important to continued student motivation [19].

2.2.4 CAL and the Internet

With an estimated 20 million web sites by the year 2000 [20], the Internet caters for a diverse and ever expanding base of Information on a multitude of topics. This information can be presented in a variety of different media, form text to sound and video. High level programming languages such as Java also allow for the development of interactive examples and demonstrations. Communication between users and lecturers is facilitated via email and other conferencing tools such as mailing list and message boards.

In this way the Internet provides a platform suitable for the development of CAL courseware but it also provides the users with a wealth of additional information upon which to draw from.

2.2.5 Learning via the Internet

The Internet has emerged as a powerful and ever expanding medium over the past decade. The web browser, introduced commercially for the first time in 1993 [4], allows educators to enrich their learning material by incorporating multimedia features. As discussed in Section 2.2.2, the hypertext/hypermedia approach allows them to design courses so as to facilitate a diverse range of learning abilities.

The Internet can cater for both synchronous and asynchronous mode of course delivery. The synchronous approach give rise to the computer classroom, with computers being provided for each student [21]. While effective it requires considerable investment in both computer hardware and lecturer time to control and monitor the sessions.

The asynchronous mode to online learning offers a more student focused approach, rather than the traditional teacher-controlled environment [22]. This mode is particularly suited to distance learning programmes. The student can work through the material at their own pace and in a manner that suits their own learning needs.

2.2.6 Accessibility and Resources

Access to the Internet is rapidly increasing. In America, surveys show phenomenal increases in the number of personal computers (PC's) accessing the Internet (approx. 45 million in the first quarter of 1998, up 43% from the same period in 1997 and up 140% from 1996) [23]. Such access provides an ideal platform for the delivery of courses with students requiring no extra software, other than a web browser, to take advantage of the immense information resources. It is equivalent to having access to a number of research libraries online and also similar access to other resources which are not normally used in an educational setting [24].

2.3 Modes of Learning

The traditional classroom environment is the most common mode for the delivery of education. However, distance or remote learning is emerging as a force within the education system. This section examines both with the intention of extracting the positive features of both for incorporating into a CAL package.

2.3.1 Traditional Classroom

Within the traditional classroom environment there is an inherent structure that students are socialized into throughout their years of schooling. There is a recognized code of behaviour that must be adhered to, and generally a pre-established time frame for lessons/lectures. The presentation of material is generally via a verbal presentation accompanied by black/white board and/or overhead projector material. Gestures and movements, on the part of the teacher, can also aid the communication of information to the students [25]. These are all physical attributes, however, there are also the student-teacher and student-student relationships that can influence the learning environment. Within the classroom, learning has a major interpersonal component through peer discussion [26]. Competition and cooperation between students are also important factors in the learning process. Often, these factors can serve to motivate students to learn, although motivation through competition, has been found to create unfavourable outcomes such as anxiety, low esteem and poor relations between students.

2.3.2 Distance Learning

> *Distance learning is an emerging paradigm where students, lecturers and equipment may be in different geographic locations* [27].

Distance education is a growing area within education. As a learning medium it can serve many purposes.

- Distance education by nature allows students to partake in educational course regardless of their geographic location. Students, who might not otherwise be able to attend a university on a full-time basis, can undertake courses part-time. In this way distance education can be viewed as a tool for *educating the masses.*

- In today's business world, and particular in the area of technology, skills are constantly changing to meet new demands. For employees who need to keep abreast of these, distance education in the form of Continuous Professional Development (CPD) provides an ideal solution. They can study courses part-time to continually update their skill base.

- On a more subtle level, distance education offers relative freedom from discrimination due to physical characteristics (or racial/cultural factors), that may arise in the traditional classroom [22]. It also provides a solution for those with physical disabilities, for whom attending and navigating a university campus would prove too difficult.

Unlike the classroom environment, the student is in control of the time and pace of their study. This requires a certain level of maturity and for this reason distance education is most suited to those at graduate level [22]. Depending on the course setup, interaction with other students is minimal and therefore students tend to miss out on the benefits this interaction can bring. To counteract this some courses provide a week long workshop where student come and work on team projects. In this way, some class interaction is introduced, however, depending on numbers and location of students this may not be practical. This ability to communicate with each other to discuss topics, as is the case in the traditional classroom, is important. Students learn that drawing on the strengths of the group to produce a better answer is much more effective than individual/competitive efforts [22].

2.3.3 Education and CAL

2.3.3.1 Distance Education and CAL

The Internet provides an exciting CAL platform upon which to deliver distance education courses. The multimedia/hypermedia capabilities of the Internet allow for the enrichment of course material, while ever increasing access make it an ideal distribution network. All the material resides on a single host server, so modification or updating of courseware involves changing only one set of documents with no need for redistribution afterwards. The existing and costly, both in terms of money and time, requirement to distribute course material by mail can be replaced by the more efficient electronic distribution system offered by the Internet.

The communication facilities of the Internet also offer considerable benefits to students. Tools such as e-mail, conferencing, bulletin boards and mailing lists greatly improve the communication between student and lecturer and also promote collaborative work among students themselves. This gives rise to the virtual classroom. This is a teaching and learning environment located within a computer-mediated communication system [28]. As discussed in Section 2.2.3, the ability of students to communicate with each other to discuss topics is important. It can develop a team spirit amongst the students and work to motivate them in their quest for knowledge.

It is these mechanisms, provided by the Internet, that make it ideal for distance based learning programmes. However, it is not merely enough to transfer existing "*class*" material to the web medium. A more comprehensive and complete course structure is necessary to cater for those in remote areas who do not have direct access to library material, etc. Often additional material can be found on the web itself but it is important to incorporate this in a structured way. In fact, the presence of unwanted material can present a problem to the web based student as it is often more interesting than the material required for the educational task being performed [24].

2.3.3.2 Traditional Classroom and CAL

As mentioned in Section 2.1, the traditional classroom can also benefit from the introduction of CAL. Computer simulations and graphical examples can be incorporated into lectures to provide students with visual explanations of material.

Additional material can also be provided so that students can further explore topics outside of the classroom. The use of computer quizzes would allow student to continually assess their level of understanding, thus giving them the opportunity to address potential problem areas.

2.4 Conclusions

Based on the ideas presented, it is the opinion of the author that the Internet and WWW present the opportunity for an exciting new era in the delivery and development of educational material. The nature of the medium allows for a variety of multimedia and interactive features, which have been shown to increase student interest and motivation, to be incorporated into the education process.

The underlying hypertext/hypermedia format of the Internet allow educators to design courses that cater for all learning styles and abilities. It also offers new possibilities in the area of distance education, by providing a platform for the development of programmes that can give the user a more comprehensive learning experience.

CAL can also enrich traditional classrooms by providing students with additional material in the form of interactive examples. Students, through the use of these examples, can explore ideas in greater depth and in a more enjoyable way.

Chapter 3

Approaches to Online CAL

3.1 Introduction

This chapter investigates some of the current approaches used to present courses and/or course material via the Internet. These can range from simply making notes available for download to fully interactive online courses. Issues such as time on the part of the lecturer, cost, and the lecturers own experience can dictate to what extent they will incorporate the medium into their teaching structure. These factors are examined and a possible incremental development process is discussed. The chapter also examines some of the commercial software available for the development of online courses.

3.2 Online Course Material

The extent to which a lecturer may incorporate web material into his/her course can vary greatly. They may decide to use the web merely to provide student and potential students with details of the course, such as a table of contents, time scales, etc. They may decide to supplement their in-class lecturers with online material or they may decide to present the course entirely online. The Internet provides the flexibility for all these modes of delivery.

3.2.1 Supplemental Online Material

Traditional classroom lectures usually involve the lecturer either elaborating on pre-written slides or overheads or using the black/white boards to present notes to students. A considerable benefit to students is to have these notes or handouts before the lecture so they can concentrate on the extra information, usually verbal on the part of the lecturer, rather than spending their time transcribing slides. The Internet provides the lecturer with several options as to how to supplement or indeed present his/her course.

3.2.1.1 Incremental Development Process

The extent to which a lecturer will incorporate online material into their course may be, proportional to their own experience with and competence in using the web. Time is also an important factor when it comes to developing and designing online material. There are many books and materials on instructional, hypertext and interface design, but educators and students may not have the time to read them all nor to decide which is most suited to their needs and ability [17].

Initially, the lecturer may use the Internet to outline his/her classroom course. A contents list may be posted along with details of any assignments or laboratory work that has to be completed. They may elect to host the course notes in Word [29] or Postscript format for students to download and print out before class. References to other reading material, both on and off line, can be included. For large classes, communication in the form of mailing lists or message boards may be incorporated. This allows students to pose questions to the lecturer or other students and have the answers made available to everyone, thus avoiding repetition of problems etc. E-mail can be used for more sensitive one to one situations.

The development and incorporation of these features can be a step by step process that can be hand crafted by the lecturer to suit his/her own course needs. More advanced features such as online tutorials or interactive Java examples can also be included but these generally require some web based programming experience to code and maintain if developed from scratch.

3.2.2 Entirely Online Course

Course that are presented entirely online are most suited to the distance learning mode of education. The growing access to the Internet means that educators can reach students whose geographic location is remote from the university. However, as described in Section 2.3.2, the communication and collaboration facilities must be in place so as not to leave the students feeling isolated. Therefore, unlike the situation for supplemental material, these features must be in place from the outset.

3.3 Commercially Available Software

The field of online education is growing at such a rate that commercial industry has begun to recognize a need for tools that will enable online course developers to create and manage their courses in a structured and user friendly way. As a result many software tools have entered the market, each aiming to take a stand in this rapidly expanding area. These include packages such as WebCT, Topclass, Lotus LearningSpace 2.0, Authorware 4 and many more. On brief inspection Learning Space 2.0 and Authorware 4 were both ruled out. LearningSpace only runs on the Lotus Domino Server which means users would have to change over to this, with its associated high cost and support overhead, just to use the package. Authorware also requires additional software, in the form of Windows based Solis Pathway, in order to implement administration and student tracking capabilities, which are essential for evaluation purposes [34]. Based on this, the author has chosen two of these commercially available packages for further evaluation, WebCT and Topclass. The features of both are outlined and evaluated in the following sections.

3.3.1 WebCT

3.3.1.1 Overview

WebCT is a tool that facilitates the creation of sophisticated World Wide Web-based educational environments [30]. It provides for the development of entire on-line courses, or supplemental material for existing courses. The interface for both students and teachers alike is via a web browser. No additional software is required to view or develop course material. The WebCT software is housed and run on a UNIX server, although a Windows NT version is currently in development [31]. It can be viewed, however, through any browser.

All material within WebCT is HTML based. This allows course designers to make use of existing converters and/or HTML editors to create course pages. These can then be uploaded to the WebCT package. WebCT itself does not cater for features such as mathematical symbols and notations, however, because it uses HTML, the package is open and transparent to upgrades in HTML standards, without any modification to the package itself.

WebCT provides a number of tools that aid the lecturer in presenting and maintaining their course. The package allows for the inclusion of online quizzes and tutorials, conferencing and email as well as features like calendars, learning goals and page annotations. Each student has their own homepage and because they have a unique login id, the package can remember the settings. For instant, when they login, it gives them the option of returning to the last page they covered in the previous session. Using the unique login ids, the package keeps track of each student progress through the course material. Management of grades and other student details is also catered for.

Essentially, the aim of WebCT is to provide a platform of facilities around which the lecturer can incorporate his/her course material. It does not offer a guide as to how the course material should be structured from an educational point of view. It is merely a tool for housing and managing the online course.

3.3.1.2 Evaluation

WebCT can be installed and run from a local server or alternatively some service providers have installed WebCT and these can provide access. It is at present restrictive in that the host server must be UNIX based, however Windows NT versions are currently be developed. WebCT does not supply its own HTML editor. It allows for the development of the HTML course page on any commercially available editors which can then be uploaded to WebCT. In terms of development, WebCT was found to be very user friendly. Figure 3.1 shows a screen shot of a sample WebCT homepage, as seen by the developer (the student view is similar except they do not see the control buttons), complete with control buttons for developing and modifying the course. WebCT was found to have a wide variety of tools and features to aid both student and lecturer with good student tracking facilities and because it is Internet based the hypertext approach to course design as described in Section 2.2.2 can be applied.

WebCT is free during the development stage of a course. Charges are incurred when student accounts are created, i.e. when the course is made available to students. Pricing then depends on the duration of the course and number of student accounts created. See Table 3.1 for pricing structure. As the number of courses increase, these figures become quite a substantial on-going expense. As with any course, development and maintenance would be necessary in terms of creating and updating course material, and when added to the annual charge of running a WebCT course, the overall cost was deemed unsuitable for the present needs of the intended neural networks course.

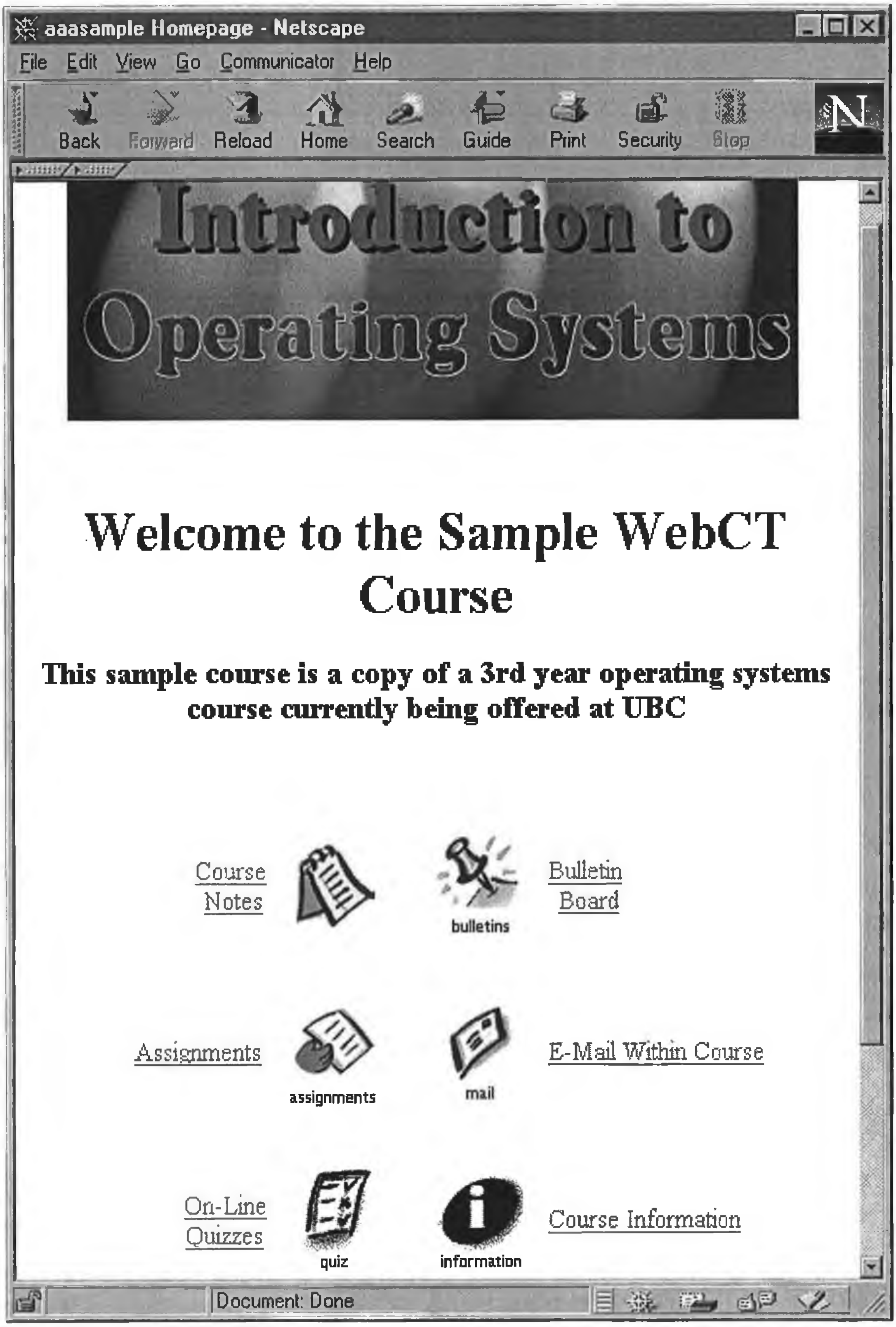


Figure 3.1: Instructor view of WebCT homepage

Number of Accounts On Server (sum of WebCT Student accounts over all courses)	License Charge For That Server			
	4 Month	6 Month	8 Month	12 Month
50	$100	$140	$180	$250
71	$150	$210	$270	$375
100	$200	$280	$360	$500
200	$300	$420	$540	$750
400	$400	$560	$720	$1000
800	$500	$700	$900	$1250
1600	$600	$840	$1080	$1500
3200	$700	$980	$1260	$1750
6400	$800	$1120	$1440	$2000
12800	$900	$1260	$1620	$2250
25600	$1000	$1400	$1800	$2500
51200	$1100	$1540	$1980	$2750
Unlimited Single Server License	$1200	$1680	$2160	$3000
Corporate License (per server)	$1200	$1680	$2160	$3000

Table 3.1: WebCT costing structure:[32] Prices correct on 27/7/98

3.3.2 Topclass

3.3.2.1 Overview

Topclass provides a virtual classroom environment to manage all aspects of content and class management [33]. Like WebCT, it provides the mechanics and facilities around which a lecturer can design his/her course. The interface is via a web browser and the software, known as Topclass Server, runs on Windows 95/NT and UNIX platforms. The package provides all the facilities of WebCT such as collaboration tools, testing and grading facilities and student and content, management. All the course material is stored in modules in database form. This allows the lecturer to reuse existing materials in other courses where appropriate.

3.3.2.2 Evaluation

Like WebCT, Topclass is Internet based and has all the benefits that this provides including good communication facilities such as email and built-in threaded discussions and the ability to incorporate multimedia features such as Java, Shockwave, or any other Web-based technology applications. An evaluation by PCWeek [34] rated Topclass in first place with a score of 98 out of 100, 1 ahead of LearningSpace 2.0 and Authorware 4. WebCT was not included in the evaluation. Figure 3.3.2.2 shows the instructor view of the Topclass environment. It uses templates to allow instructors to create course with minimum programming experience. However, like WebCT, as all material is HTML based, there is no provision made for incorporating mathematical symbols or equations. Both rely on improvements in the HTML standard for these features. From this point of view they offered no real benefit over standard web publishing facilities.

Topclass comes with a yearly license fee. The exact cost is determined by the number of users. Table 3.2 shows a breakdown of the cost (in US dollars).

No. of Users	License Lease
25	$1450
50	$2250
100	$3750
200	$4750

Table 3.2: TopClass costing structure: Prices correct on 3/9/98

As was the case with WebCT, this ongoing cost was deemed unsuitable for the needs of the ANNet course at this present stage.

Figure 3.2: Instructor view of Topclass homepage

3.4 A Proposed Structure for an Online Course

In order to investigate the effectiveness of the Internet as an environment for the development and delivery on educational courses, the author proposes the development of an online course to teach the subject of Artificial Neural Networks. Having decided against the use of a commercially available package for this undertaking, the author proposes to develop an Internet based hypertext environment, based on the discussion in Chapter 2, that will incorporate interactive features to aid the understanding and overall learning experience of those who undertake it.

Figure 3.3 shows, in diagram form, the components that this hypertext environment will incorporate.

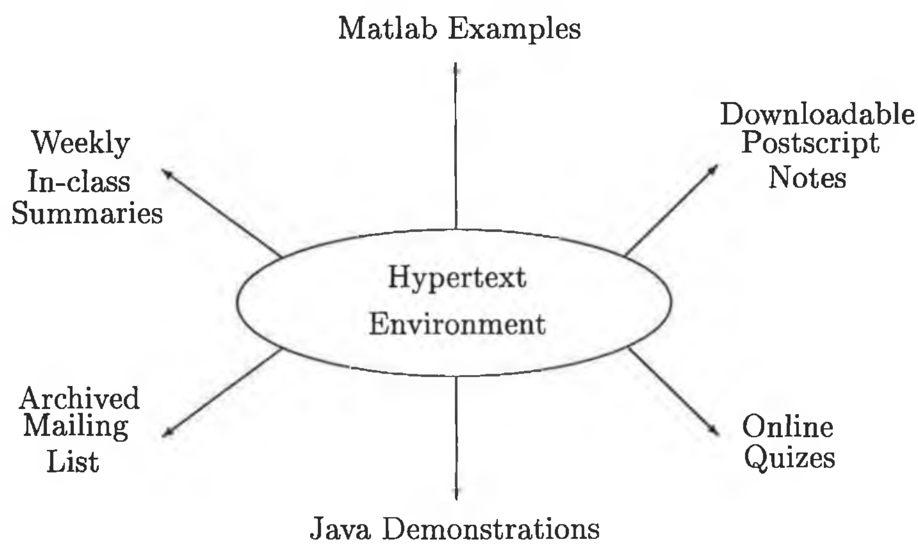


Figure 3.3: Proposed Course Structure

The course will be broken up into *"chapters"*, based on the material content. Students of all learning abilities will be catered for, with hyperlinks providing further explanations for those who require them. At the end of each *"chapter"*, instant feedback online quizzes will give students the opportunity to evaluate their understanding of the material they have just covered. References to further information, both online and in printed form, will be provided where applicable throughout the material. Java demonstration and MATLAB™ examples will allow the student to explore and examine the concepts behind the various topics covered. The students will be able to change parameters and settings and view, in graphical form, the outcome of their action.

An archived mailing list will allow the lecturer to communicate with students regarding class issues while each student will have an individual email address. To allow students taking the course remotely to pace themselves a weekly summary will be posted via the mailing list to all pupils and will detail the progress and topics covered in class.

To help evaluate the usage of the course by students a tracking system will operate in the background. This will log the movement of the students through the course material. Feedback from the online quizzes will also be monitored to allow the lecturer to expand material in areas that are posing problems to the majority of the class. This is important for evaluating the effectiveness of the instructional material [17].

Chapter 4

ANNet :- An Artificial Neural Networks CAL Package

4.1 Introduction

This chapter will introduce the relatively new and ever expanding area of Artificial Neural Networks (ANNs). It will provide the motivation for both the subject itself and also the reason for its inclusion and suitability for a remotely taught course. The author will then proceed to give an introduction to the field of ANNs. The chapter will conclude with a description of and motivation for the features and components of the ANNet Artificial Neural Networks course.

4.2 Motivation for ANNs

ANNs are a relatively new area with essentially three main areas of expertise, classification, function approximation and associative memory. They are massively parallel networks and have the ability to generalise (provide reasonably good outputs between known points, discussed later in chapter). Because of this they have found their way into many applications both in the scientific/engineering domain and more recently in the business world.

The classification abilities of ANNs have found them uses in areas such as fingerprint and handwriting recognition [35], image processing [36] [37], and various diagnostics systems (including applications in the medical field) [42]. The ANN can be trained to recognise combinations of symptoms as well as qualitative information such as time of year, patients environment, etc and based on these, decide what the most likely diagnosis is.

Function approximation has found many useful applications in the area of forecasting. When used properly, they can be trained to forecast a wide variety of phenomena from Sun Spot Series [43] to financial market trends [44]. These type of ANNs also find applications in areas such as system identification and modeling [45] and control [46].

As demonstrated, ANNs have found a wide variety of applications throughout many different disciplines and through research and development many more could arise, thus, making the field of ANNs a worthwhile consideration when trying to solve many problems.

4.3 Motivation for ANNs as a Remotely Taught Course

As outlined in Section 4.2, ANNs have found many applications through industry. However, as they are a relatively new field, many people within industry would not have the technical understanding of how they operate. This is due to the fact that ANNs are not usually taught as an undergraduate subject, they are generally found in postgraduate or Masters courses.

This leaves a wide open market to teach this area of expertise to those who require it. Because many of these people would be in full-time industry they generally don't have the time to take full-time courses and so part-time or Continuous Professional Development (CPD) courses are ideally suited to their needs. Through this form of education they can study at times that suit them and in the case of CPD they don't necessarily have to sit exams at the end. They can use the structure of a taught course as a convenient and comprehensive way of learning about the subject.

As discussed in Chapter 2, the Internet provides an ideal platform upon which to offer these courses, giving the user widespread access to not only the course material but also to the plethora of information hosted on the web.

4.4 Artificial Neural Networks

4.4.1 Introduction

Work on ANNs , commonly referred to as *"neural networks"*, has been motivated right from the beginning by the recognition that the brain computes in an entirely different way from the conventional digital computer [7]. They consist of highly interconnected processing units, called neurons, that can respond to input stimuli and learn to recognise and adapt these inputs to produce a desired output.

The inspiration for these artificial neurons comes from the brains own processing units also called neurons. These biological neurons consist of:

- A cell nucleus, which is the body of the neuron,
- A set of dendrites which receive the neurons inputs. These are usually the outputs from other neurons, thus giving the brain its highly connected structure.
- An axon which is the output of the neurons. Through synaptic junctions, the axon can branch out at its extremities to act as inputs to several other neurons.

Figure 4.1 shows in diagram form the structure of one such biological neuron. The neuron operates by summing up its inputs and if they exceed a certain threshold value then the neuron *fires*.

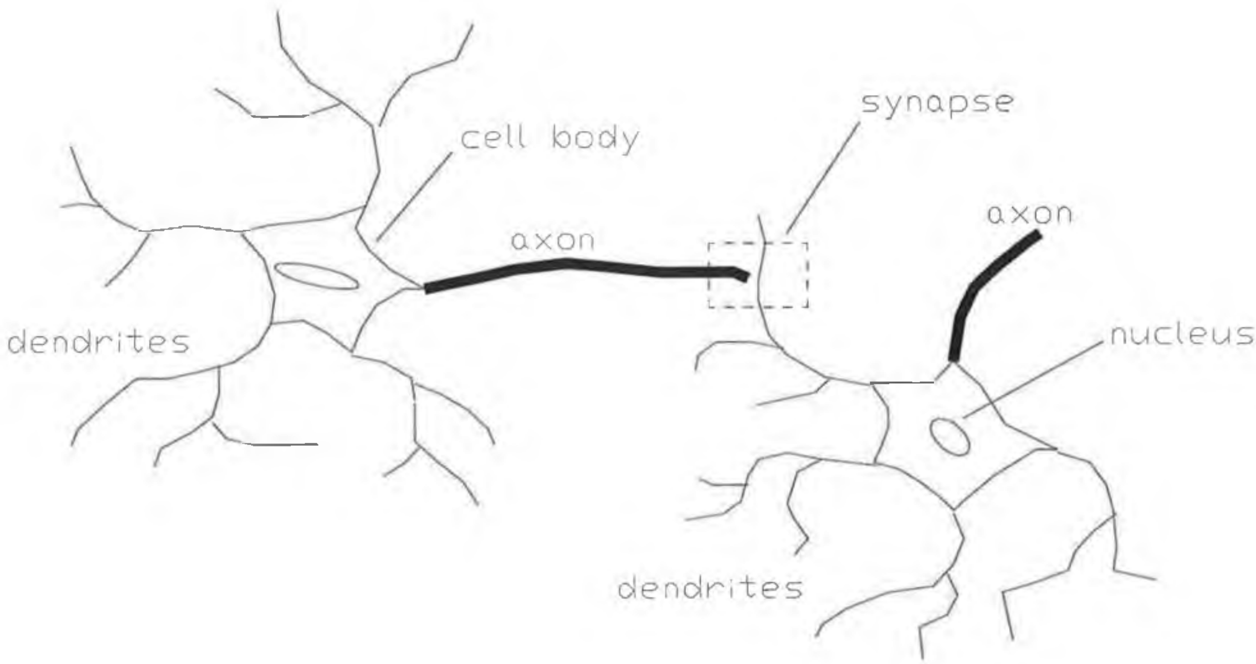


Figure 4.1: Schematic Drawing of Biological Neuron [38]

Artificial neurons attempt to mimic this mode of operation. A model of such a neuron can be seen in Figure 4.2. In this neuron the inputs, x_i can be considered as the outputs from previous neurons, i.e. in the biological sense other neurons axons. They can however, be external inputs in the case of ANNs. The inputs are then weighted via a set of weights w_i and summed for processing by the neurons activation function. These activation functions may take any form but, in general, ANNs use either sigmoidal, or hard limiting type functions. Normally, the output range of the neuron is on the closed intervals, [0,1] or [-1,1]. A positive or negative bias is generally added to lower or raise the net input of the activation function [7].

These neurons are the basic building blocks for ANNs. There are several types of ANN architecture or topologies. These are characterised by the way the neurons are interconnected with each other. As will be outlined later in the chapter, different topologies are suited to solving different types of problems.

Probably the most significant property of ANNs is their ability to *learn* from their environment, and to improve their performance through learning. Learning takes place through an iterative process of adjustments applied to certain parameters of the ANN, notably the weights [7]. In order to learn the network must be *trained*

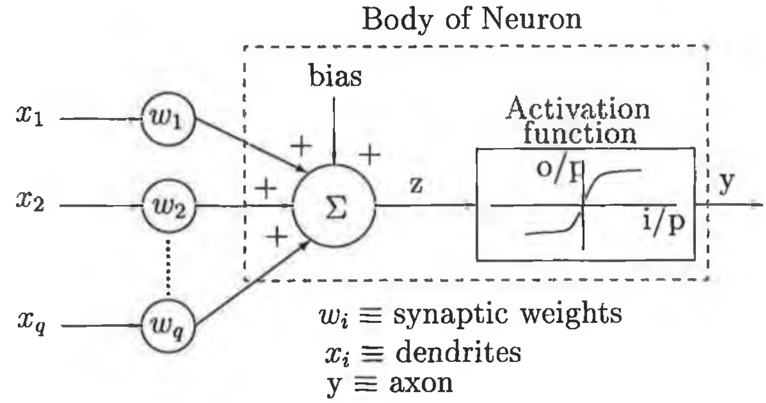


Figure 4.2: A model of an artificial neuron [38]

using a set of known input-output patterns. Once trained, the ANN does not merely reproduce the patterns it has learned. It also has the ability to *generalise* (i.e. can sensibly interpolate input patterns that are new (other than those used during training) to the network [6]).

4.4.2 Simple Neural Networks

4.4.2.1 The McColloch-Pitts Neuron

The McColloch-Pitts neuron [39] is perhaps the earliest artificial neuron [8]. They represented the activity of individual biological neurons using simple threshold logic elements and showed how networks made out of many of these units interconnected could perform arbitrary logical operations [47]. This model now known as the McColloch-Pitts neuron is shown in Figure 4.3. The inputs $x_1, x_2, \ldots, x_q$ are 0 or 1 depending on the absence or presence of the input impulse [6].

The sum of the inputs is fed in the activation function, which in this model, is a *hard-limiting* function. The output of the neuron is forced to either 0 or 1 depending on the input value to the hard-limiter. If the input is less than zero the output is 0 and if the input is greater than or equal to zero then the output is 1.

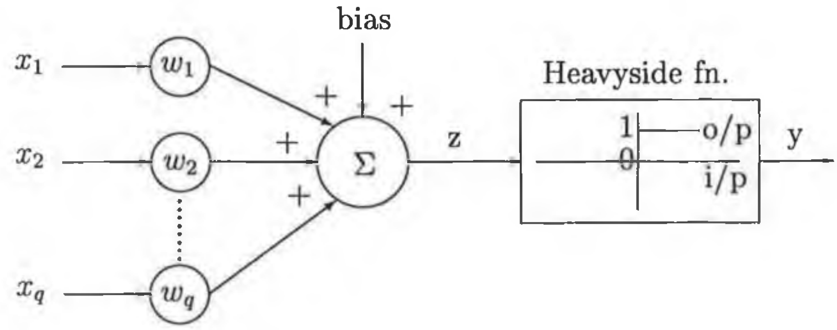


Figure 4.3: McColloch-Pitts Neuron

4.4.2.2 Hebbian Learning

One of the earliest learning rules for updating the weights of a neuron was that of Donald Hebb in 1949. He proposed that

> "When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes place in firing it, some growth process or metabolic change takes place in one or both cells, such that A's efficiency, as one of the cells firing B, is increased" [40].

Read simply, this learning rule states that if the correlation of the output/input term i.e. the crossproduct of $y_i x_i$ is positive then this results in an increase in the weight w_{ji}, otherwise the weight is decreased. Equation (4.1) shows the Hebbian learning rule, where μ is the *learning rate*. As can be seen the strength of the output is increased in turn for each input presented Equation (4.2). Thus the more frequent the input pattern the stronger the neurons weight vector becomes.

$$\Delta w_i = \mu y_i \tag{4.1}$$

where the output y_i is given by:

$$y_i = f(w_i^T x_i) \tag{4.2}$$

w_i is the ith weight of the weight vector W, given by

$$W = [w_1, w_2, \dots w_j] \quad , \quad j = 1, 2, \dots, N \tag{4.3}$$

and x_i is the ith input of the input vector X, given by

$$X = [x_1, x_2, \ldots x_j] \quad , \quad j = 1, 2, \ldots, N \tag{4.4}$$

with N being the number of inputs.

The new value of the weight w_{k+1} then becomes

$$w_{k+1} = w_k + \Delta w \tag{4.5}$$

For a large number of input patterns, the number of calculations also increase making it more suited to a computer environment for simulation.

4.4.2.3 The Perceptron and the Perceptron Learning Rule

The perceptron was invented by Frank Rosenblatt in 1958 [41]. It is similar in form to the McColloch-Pitts neuron with one important distinction, the activation function, which is dependent upon the application. The continuous activation function usually used by the perceptron is of the form:

$$y = \frac{2}{1 + e^{-\beta z}} - 1 \tag{4.6}$$

The output range of this function is restricted to $-1 < y < 1$. It is differentiable and so gradient search algorithms (discussed in Section 4.4.2.4) can be used to adjust the weights.

The perceptron is the simplest form of a neural network that can be used to classify a *linearly separable* problem (linearly separable patterns are patterns that lie on opposite sides of a hyperplane). The *perceptron convergence theorem* derived by Rosenblatt shows that if the perceptron is trained with data from two linearly separable classes then the algorithm will converge, thus finding a dividing or classification line between the two sets of data [6]. Figure 4.4 shows a model of the perceptron with the continuous activation function described in Equation (4.6).

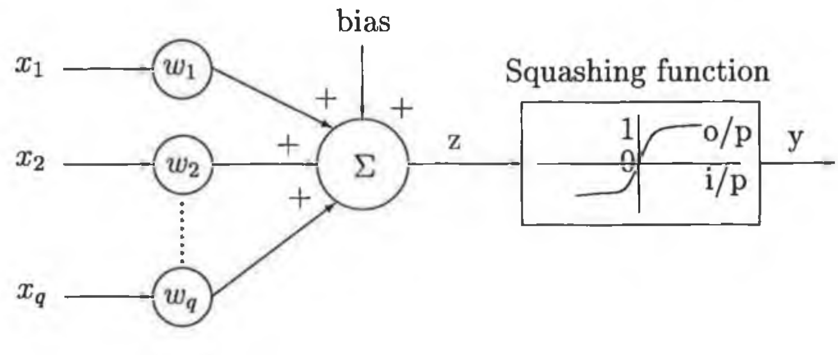


Figure 4.4: A Model of the Perceptron

The perceptron learning rule is a *supervised* learning rule and so requires a desired value of the output at each of the training points in order to train the network correctly. The rule compares the actual network output, y to the desired value, d and based on this, generates an error value, e, as shown in Equation (4.7). This error value is then used to calculate a weight adjustment, Equation (4.8). The weight update is similar to that of the Hebbian learning rule.

$$e = d - y \tag{4.7}$$

$$\Delta w = \mu e x \tag{4.8}$$

$$w_{k+1} = w_k + \Delta w \tag{4.9}$$

The perceptron learning rule is applicable to threshold type neurons (uni or bipolar in nature). As can be seen this rule is iterative, indeed the perceptron convergence theorem relies on this in order to converge to a solution, thus making it suitable to implementation in a computer environment.

4.4.2.4 Gradient Search Algorithms

The objective of network training is to find the optimum weights to minimise the error between the desired and actual response of the network. One criterion used is that of the mean-square error (MSE). When this MSE is plotted against the weights the result is a hyper-paraboloid surface, known as the *performance surface.* Figure 4.5 shows what this surface might look like for a two weight network.

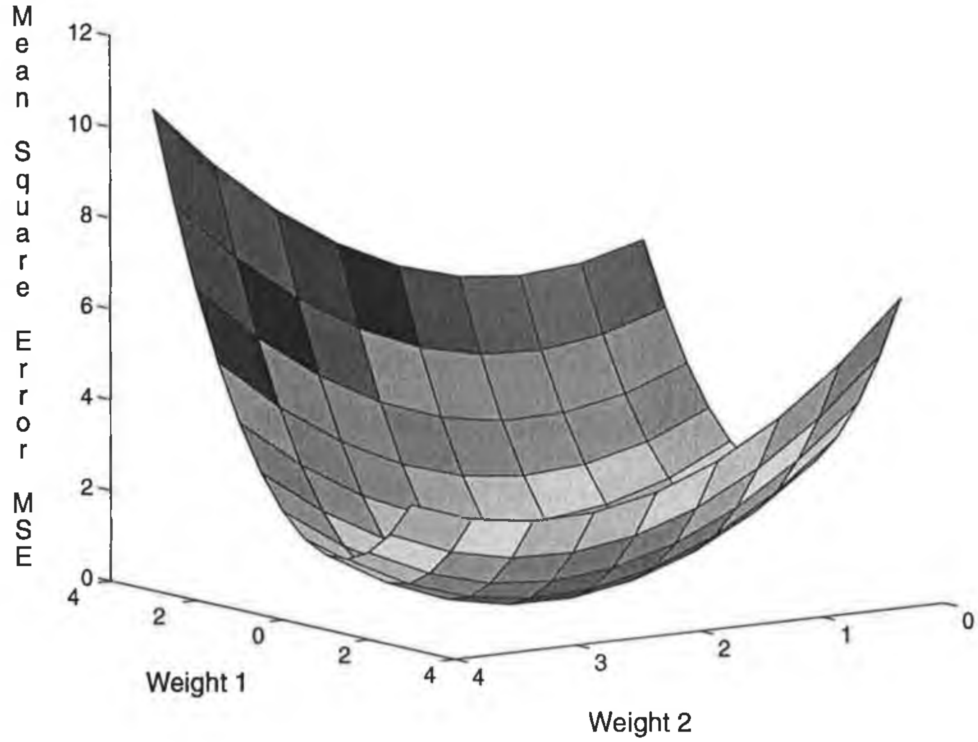


Figure 4.5: Performance surface for a network with two weights

Gradient based methods seek to find the set of weights that correspond to a minimum value for this MSE. It achieves this by making the new value of the weight a function of both the previous weight value and also the change in gradient. In this way the weights move about the performance surface to find the value of minimum MSE.

One such learning rule is the Delta Learning Rule which is described by the following weight update equations.

$$\Delta w = -\mu \nabla \tag{4.10}$$

$$w_{k+1} = w_k + 2\mu e f'(z) x \tag{4.11}$$

where ∇, the gradient is given by:

$$\nabla = 2 e f'(z) x \tag{4.12}$$

$f'(z)$ is the derivative of the function $f(W^T X)$.

One of the benefits of the delta learning rule over others such as the perceptron or Hebbian learning rule is that it can be generalised to multi-layer networks. This approach is quite computationally intensive as the gradient term has to be calculated on each iteration. Indeed to try and display such a process without the use of a computer would prove very difficult and the benefit to students would be minimal.

4.4.3 Multi-Layer Perceptron (MLP) Networks

4.4.3.1 Introduction

Perceptrons with continuous, differentiable, non-linear activation functions are the building blocks for Multi-Layer Perceptron or MLP networks. They consist of a set of input nodes, one or more hidden layers of computation neurons and an output layer of computation neurons. Figure 4.6 shows in diagram form the topology described. The input is propagated in a forward direction, through the network, layer by layer.

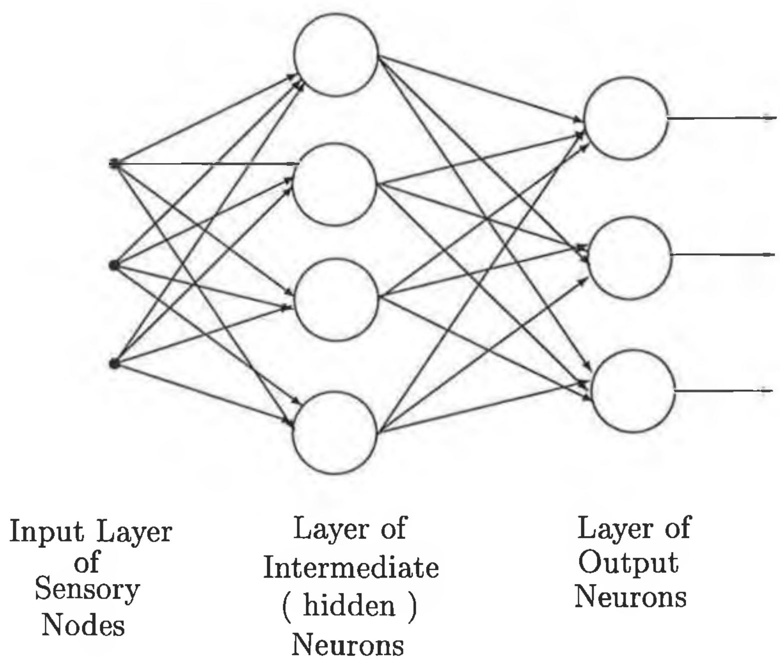


Figure 4.6: A Multi-Layer Perceptron Network

Whereas single-layer perceptron networks can solve linearly separable problems, they have difficulty with non-linearly separable ones. For example, the delta rule would generate mean-squared error solution to the problem, which may leave one or more patterns misclassified. As the number of layers is increased within the network structure, it can divide up the pattern space in more complex ways as shown in Figure 4.7 [48].

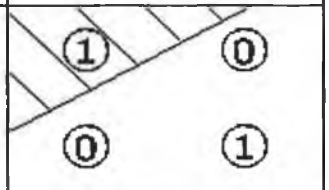
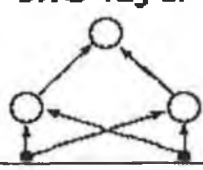
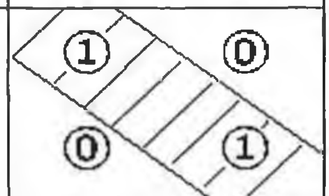
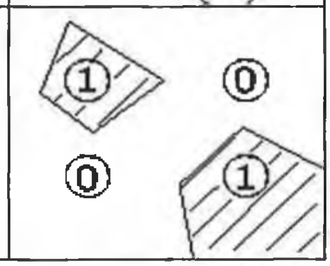
Structure	Types of decision regions	Exclusive OR problem
single-layer 	half-plane bounded by hyperplane	
two-layer 	convex open or closed regions	
three-layer 	arbitrary complexity limited by number of neurons	

Figure 4.7: ANN classification as no. of layers increase [48]

While the MLPs have been found to have clear advantages over single layer ones, there are still a number of issues which must be considered. While Cybenko's theorem shows that an MLP with just a single layer can approximate any continuous function to an arbitrary degree of accuracy, it doesn't provide any information about how many neurons should be used. Indeed, in some circumstances using two or more hidden layers, as opposed to one, may be more efficient. This problem aside, MLPs themselves were not generally used from their inception because effective training algorithms were not available [48]. However, the development of the *error backpropagation training algorithm*, which was proposed by Werbos [49] and further developed by Rumelhart and McClelland [50], has changed that.

4.4.3.2 The Error Back-propagation Training Algorithm

The error back-propagation algorithm [49][50] has two stages during the training of an MLP network. During the first stage, the forward pass, an input is applied to the network and allowed to propagate through each of the layers to the output. This is the actual response of the network to the input stimuli. During this phase all the weights of the network are fixed. During the second stage, the backward pass, an error signal is calculated at the output stage of the network by subtracting the actual network response from the desired response. This error signal is then propagated back through the network and the network weights are adjusted so as to make the actual network output move towards the desired output.

Once an error signal has been generated at the output, it is then used to adjust the weights of the neurons in the previous layer. To do this, the delta rule, described in Section 4.4.2.4, is used. The algorithm recursively calculates the local gradient at each neuron and uses this to update the weights. It is important to realize that the neurons in the hidden layers are interconnected to all the neurons in the layer ahead of it. Because of this, the error from all of these must be considered when updating the weights. The proof as described in Haykin [7], yields us with the following weight update.

$$\Delta w_{ji} = 2\mu\delta_j y_i \tag{4.13}$$

where δ_j is the local gradient across neuron j.

If neuron j is in the output layer then this local gradient is given by

$$\delta_j = f_j'(z_j)e_j \tag{4.14}$$

If neuron j is in the output layer, then δ_j is given by

$$\delta_j = f_j'(z_j)\sum_{k=1}^{M}\delta_k w_{kj} \tag{4.15}$$

Generally, the error performance surface may not be as smooth as is the case in Figure 4.5. The surface may have several local minima, some large enough to trap the algorithm, which have a larger MSE than is required. To overcome this problem there are several techniques available to try and aid the MLP to find the lowest MSE and avoid settling in these local troughs. These include adding a *momentum term* to the weight update. This involves adding a fraction of the most recent update to the current one. This re-enforces the strength of the weight update in a particular direction, giving it "momentum" to get out of and pass local minima. Another approach is to use an *adaptive learning rate*, μ, as opposed to keeping it fixed. This works by progressively increasing μ if the error is consistently reduced and dramatically decreasing it should the error grow.

The algorithm comes to end once either the number of training cycles (epochs) has been reached, or the desired MSE has been achieved, however, the network may be retrained if a desired error goal has not been reached. Depending on the application, the designer of the network may be happy to settle for a local minimum so long as his/her error criterion has been achieved.

As described, the implementation of an algorithm such as the error backpropagation algorithm requires a considerable amount of computation. Gradient terms have to be calculated across all the neurons in the network. Without the aid of a computer this would be very tedious and of little educational benefit.

4.4.4 Radial Basis Function (RBF) Networks

4.4.4.1 Introduction

Like MLP networks Radial Basis Function (RBF) networks can solve both classification and function approximation problems. Much of the early work on radial basis functions was carried out by Powell [51]. Broomhead and Lowe [52] were the first to introduce them in the design of neural networks.

RBF networks consist of a single hidden layer as shown in Figure 4.8. In its most basic form, the RBF network consists of three entirely different layers. The input layer is made up of input nodes. The second layer is a hidden layer of high enough dimension, which serves a different purpose from that in the multilayer perceptron. The output layer supplies the response of the network to the activation patterns applied to the input layer. The mapping of the input space to the hidden space is non-linear while the mapping from the hidden space to the output space is a linear one [7]. Another important distinction from the MLP topology is that the weights of the network are on the output side of the neurons. They are adapted to determine the mix of the basis functions in the networks output.

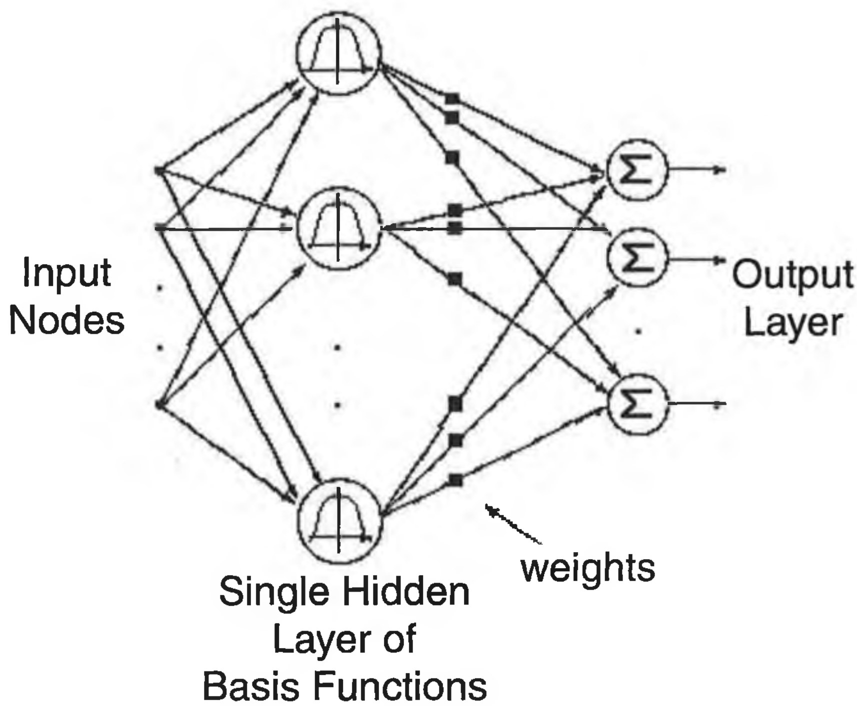


Figure 4.8: A Radial Basis Function network

A typical example of the type of basis function used is a Gaussian function. Equation (4.16) shows a two input Gaussian function which when plotted would appear as in Figure 4.9. The x_i and σ_i terms represent the *centres* and *spread factor* of the function respectively.

$$G(x; x_i) = exp\left(-\frac{1}{2\sigma_i^2}\left\|x - x_i\right\|^2\right) \tag{4.16}$$

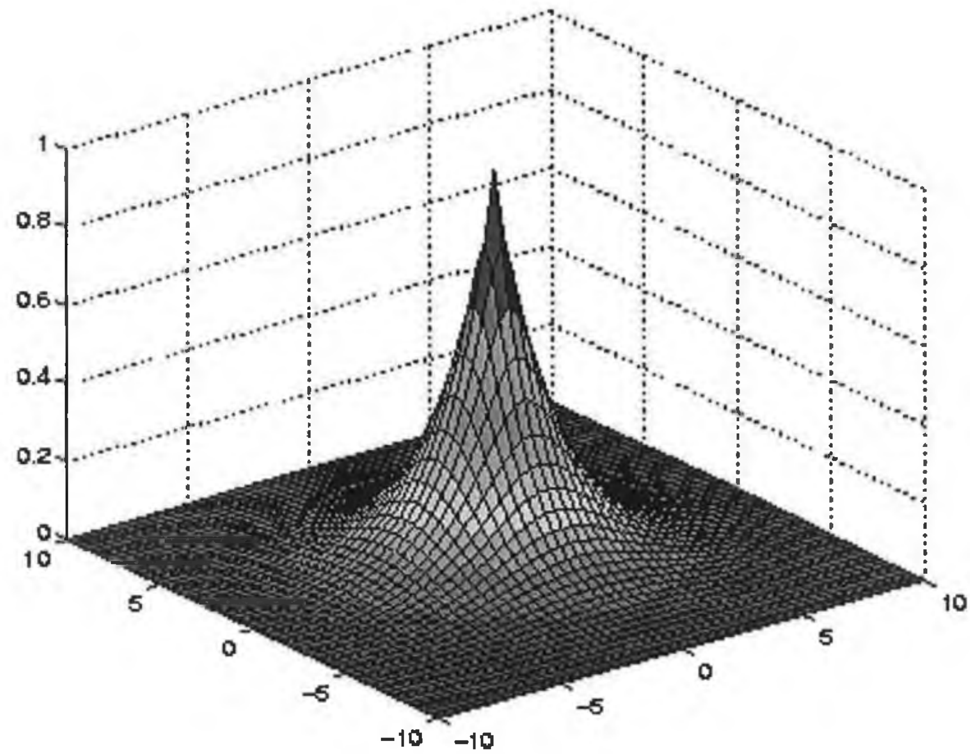


Figure 4.9: Two input Gaussian Function [38]

4.4.4.2 Functionality of RBF Network

Like MLP networks, Radial Basis Function (RBF) networks can solve both classification and function approximation problems. The primary difference between MLP and RBF networks is the nature of their basis functions. For the MLP network, the hidden layer neurons use sigmoidal basis functions which are spread over the entire input space. RBF networks use radial basis functions which are localised to only small regions of the input space.

While some problems are more suited to radial basis functions others can be better solved using sigmoidal ones (e.g. problems where the curse of dimensionality becomes a factor). RBF networks are more efficient in classification problems than MLP networks [53]. This is backed up by Covers [54] theorem which states that if a non-linearly separable problem is cast into a higher dimensional nonlinear space then it is more likely to be linearly separable. RBF networks, however, do suffer from what is known as the *curse of dimensionality.* As the dimension of the input space increases, the number of neurons grows exponentially. This can limit the use of RBF networks in certain applications.

4.4.4.3 RBF for Function Approximation

In the case of function approximation problems, the network seeks to map the input space to the higher dimension space and arrange the basis functions in such a way that the centers of these functions are positioned at each of the training data points. Essentially, they seek to find a function, F(x), such that when it is applied to each of the training points, x_i, it will give a desired output, d_i. RBFs are therefore trained in a supervised manner. Mathematically the problem can be defined as:

$$F(x_i) = d_i \quad , \quad i = 1, 2, \ldots, N \tag{4.17}$$

In order to find the value of the function in between the training points, interpolation is required. The interpolating function specified by the RBF network is given in Equation (4.18).

$$F(x) = \sum_{i=1}^{N} w_i \varphi(\|x - x_i\|) \tag{4.18}$$

It can be shown [7][38] that when solving for the weights, the matrix equation

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{N1} & \varphi_{N2} & \cdots & \varphi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \tag{4.19}$$

where

$$\varphi_{ji} = \varphi(\|x_j - x_i\|) \quad , \quad j, i = 1, 2, \ldots, N \tag{4.20}$$

is obtained. Φ is a matrix of the outputs of each of the φ functions when they are applied to the data points and d is the desired outputs at each of the training points.

This may be written as:

$$\Phi w = d \tag{4.21}$$

In order to solve for w, Φ must be positive definite (i.e. its determinant must positive and non-zero). This guarantees that Φ is invertible.

One of the main problems here, is that a neuron is created for each training point. This can lead to poor generalisation (interpolation between these points). In order to improve the generalisation of the network a complexity cost is taken into consideration. Now the overall cost becomes a trade off between matching the training data points and the complexity of the network. It can be shown that this can be achieved by adding a factor ρ, called the *regularisation parameter*, times the Identity matrix, I to Φ (Φ is the matrix of outputs from the functions that place a centre at each of the data points). The result is that Φ becomes more diagonally dominant and so improves the conditioning of the matrix. (in other words the determinant of the matrix becomes large and so small changes in the determinant don't have a dramatic effect). If this ρ factor is zero then we are back at the original case of Equation (4.21).

To help improve the performance of an RBF network, the number of neurons can be reduced. This reduces the amount of computation and also the complexity of the network. However, it does pose a new problem, that of where to position the centres. To overcome this various learning strategies can be applied, the simplest of which is the *fixed centres* approach. This involves spreading the centres evenly over the input space. The spread factor is now chosen according to:

$$\sigma = \frac{l}{\sqrt{2M}} \tag{4.22}$$

where l is the maximum distance between centres and M is the number of centres

As is the case with MLP networks, RBFs require considerable computation. This is demonstrated by equations such as Equation (4.18) and (4.19), which lend themselves to a computer-aided solution.

4.4.5 Hopfield Networks

4.4.5.1 Introduction

ANN architecture can be roughly divided into two main types, the *non-linear feed-forward networks* and the *recurrent networks* [7]. MLP and RBF networks belong to the former while Hopfield networks belong to the latter. Hopfield networks [55] were proposed by John Hopfield in 1982. They can be viewed as a non-linear associative memory or content-addressable memory devices. One of the Hopfield networks original applications was associative memory. An associative memory is a device which accepts an input pattern and produces as an output the stored pattern which is most closely associated with the input [53]. This input is usually an incomplete or noisy version of one of the stored patterns, and is in binary form. A model of the structure of such a network is shown in Figure 4.10. Hopfield networks can be either continuous or discrete-time. They are recurrent networks because the output of each neuron is fed back to the inputs of all the other neurons.

An important property of a content-addressable memory is its ability to retrieve a stored pattern, given a reasonable subset of the information content of that pattern. The stable points of the phase space of the network represent the fundamental memories of the network, i.e. the locations of these stored patterns. A noisy or incomplete version of one of these fundamental memories would have a particular starting point on this phase space depending on how closely it resembles the original. Providing this starting point is close to the stable point of the pattern memory being retrieved, then the system should converge, over time to the stable point. At this point the entire memory is reproduced by the network [7]. These stable points occur at the local minima of the energy function of the network which is given by [7].

$$H = -\frac{1}{2}\sum_{i,j} w_{ji}\zeta_j\zeta_i \tag{4.23}$$

where ζ_i represent all the possible states of the system.

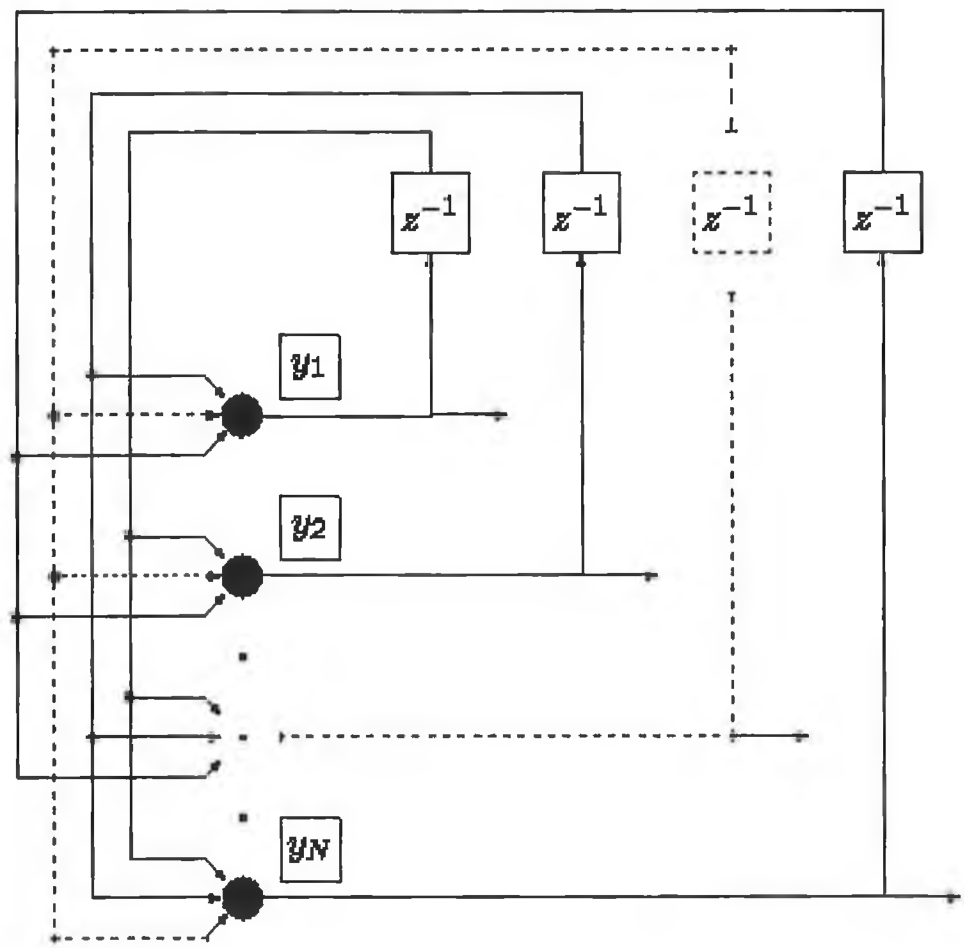


Figure 4.10: A Hopfield network [38]

4.4.5.2 Operation of Hopfield Network

As both the inputs and outputs of the Hopfield network are binary, the neurons are very similar to the McColloch-Pitts threshold neurons [39] described in Section 4.4.2.1, except for one feature. Instead of just outputting +1 or -1 for y_{k+1}, they can also output the previous output y_k if $z_{k+1} = 0$ (z is the input to the neurons activation function i.e. the sum of the inputs times the weights). In this way there has to be some memory to store this value should the $z_{k+1} = 0$ condition occur. There are two methods of updating the neuron outputs, either synchronously or asynchronously. Synchronously means that all the outputs are fed back together and all the new outputs are generated simultaneously. This requires a central clock.

With the asynchronous method, only one output at a time is changed.

In order to determine how close the stored pattern is to the current state of the network the network uses the *Hamming distance formula*, Equation (4.24).

$$d_H(\xi, \zeta) = \frac{1}{2}\sum_i |\xi_i - \zeta_i| \tag{4.24}$$

where $d_H(\xi, \zeta)$ is the Hamming distance, ξ_i is the stored pattern of neuron i and ζ_i denotes the current state of neuron i. This Hamming distance can be interpreted as the number of bits which are different between ξ and ζ.

An examination of the stability of a pattern yields a crosstalk term (i.e. another pattern that interferes with the pattern to be stored). It is this crosstalk term that affects the stability of the network. For a particular pattern, ζ_j^v, the crosstalk term is given by

$$C_j^\nu = \frac{1}{N}\sum_i \sum_{\mu \neq \nu} \xi_j^\mu \xi_i^\mu \xi_i^\nu \tag{4.25}$$

If it is positive and its magnitude becomes greater than 1, then it can change the sign of z_j, the input to the neurons activation function, and so cause an erroneous pattern to be outputted. As the number of patterns increase so too does the crosstalk term. The result is that the basins of attraction decrease in size and so there is a greater probability for error. Thus the number of states that can stored to give us a certain percentage of correctness can be determined. This value depends on, N, the number of bits in the pattern. Thus the maximum number of patterns, of N bits, that we can store, p_{max} is given by,

$$p_{max} = \frac{N}{4log(N)} \tag{4.26}$$

For example if N is equal to 80, then the maximum number of patterns that can successfully be stored is 4, ($p_{max} = 4.56$) This assumes that the patterns are randomized. If they are not this value decreases. There are some ways of increasing this bound by preprocessing the data of the patterns. For example, if the patterns are orthogonal there is no crosstalk and so theoretically, N patterns could be stored. Overall it is this limitation on the number of possible stored states that limits the applications of Hopfield networks.

As described, the Hopfield network operates by receiving an input in the form of an initial condition on the state. It is then allowed to recursively iterate until it find the pattern it deems to be the correct output. This iterative process lends itself well to a computer simulation environment.

4.4.6 Kohonen Self-Organising Feature Maps

4.4.6.1 Introduction

The principal goal of the self-organising feature map developed by Kohonen [56], in early 1981, is to transform an incoming signal pattern of arbitrary dimension into a one- or two-dimensional discrete map and to perform this transformation adaptively in a topologically ordered fashion [7]. As suggested by the name, the network develops a *map* of the input space with particular neurons responding to inputs from specific regions of this input space. This is somewhat similar to the internal representation of information in the brain. Studies have shown, as reviewed in [57], that this information storage in the brain is generally organised spatially. Clusters of neurons respond together to stimuli from the parts of the body they service. For example, neurons in the visual cortex respond to certain light patterns falling on the retina [58]. Other areas are responsible for other functions such as hearing (auditory cortex), skin surface (somatotopic map), etc.

The self organising feature map (SOFM) is a simplified model of the feature-to-localised-region mapping of the brain [58]. It is an unsupervised, competitive, self organising network. The neurons within a SOFM are organised into a lattice like structure usually one or two dimensional in nature. Each input is connected to every neuron in the lattice structure. Through a competitive learning process particular neurons become tuned to various input patterns or classes of patterns. The locations of the *winning neurons* tend to become ordered with respect to each other. In other words the neurons arrange themselves spatially within the network lattice to form a map of the input space.

This self-organisation with SOFM networks comes about through a process of *competitive learning*. The neurons compete against themselves to be activated; the neuron with the largest output being the *winner.*

4.4.6.2 The SOFM Algorithm

Consider the two-dimensional lattice structure shown in Figure 4.11. The inputs to the network are given by

$$X = [x_1, x_2, \ldots, x_q]^T \tag{4.27}$$

where q is the number of inputs to each neuron in the network

The weight vector associated with neuron j is given by

$$W_j = [w_{j1}, w_{j2}, \ldots, w_{jq}]^T, \quad j = 1, 2, \ldots, N \tag{4.28}$$

where N is the number of neurons in the network

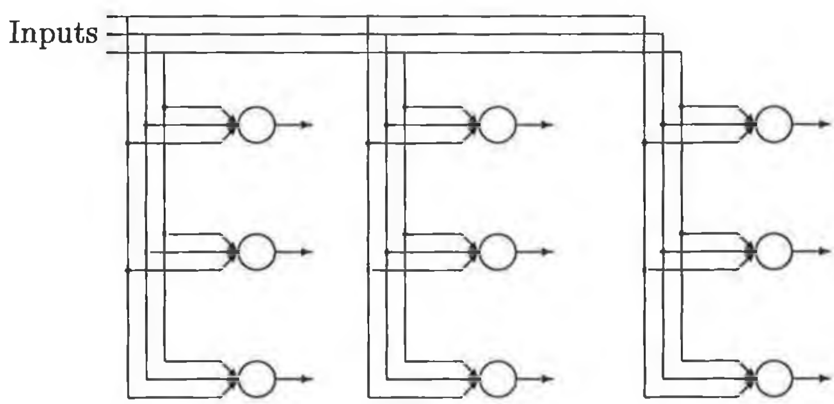


Figure 4.11: 2D Lattice of Neurons [7]

The *winning* neuron under competitive learning is denoted by the neuron with the largest output. To determine which neuron this is, the inner product can be used. These inner products $W_j^T X$ for $j = 1, 2, \ldots, N$ are calculated and the largest is selected, assuming that the same threshold is applied to all neurons. The weights of the winning neuron are then updated to move them closer to input position. The other neurons are not altered. If, however, the self-organising algorithm is to be used for, say, natural signal patterns relating to metric vector spaces, a better matching criterion may be used, based on the *Euclidean distances* between the weights and the inputs [56].

It is crucial to the formation of *ordered* maps that the cells (neurons) doing the learning are not affected independently of each other, but as topologically related subsets, on each of which a similar kind of correction is imposed. As the learning evolves, these subsets will encompass different cells. Thus the net correction of each cell will be smoothed out in the long run. The weight vectors themselves, will tend to attain values that are ordered along the axes of the network [56].

If a *neighborhood set,* N_c is defined and centred around each winning cell as in Figure 4.12, then all the cells within this neighborhood are updated as well as the winning neuron. The width or radius of the N_c can be time-varying. It has been shown, [56], that it is advantageous to make N_c very wide at the beginning at let it shrink monotonically with time. The process usually ends with N_c equal to just a single cell.

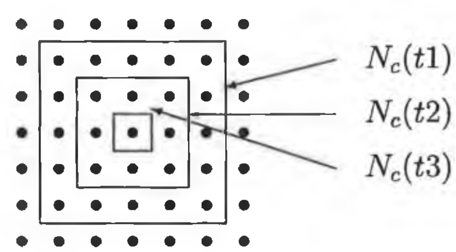


Figure 4.12: Examples of topological neighborhood $N_c(t)$, where $t1 < t2 < t3$ [56]

Now the update algorithm for the weights may be defined as

$$w_i(k+1) = \begin{cases} w_i(k) + \alpha(k)[x(k) - w_i(k)] & \text{if } i \in N_c(k) \\ w_i(k) & i \notin N_c(k) \end{cases} \tag{4.29}$$

where α is the adaptation gain, with $0 < \alpha < 1$.

The ability of the SOFM network to create a map of the input space can be used to gain an approximation of this input space. The basis for this is in vector quantization theory, the motivation for which is dimensionality reduction or data compression [59]. The input space can be broken up into regions and a single quantized value be allocated to all inputs within that region by the network.

As with all the other ANN architecture outlined, computers are essential for both understanding and, indeed, training of the network. The SOFM network like the others uses iterative techniques to converge upon a solution and so a computer environment is essential.

4.4.7 Application of ANNs

Having made the decision to use an ANN to solve a particular problem, there are several other factors that must be considered, an important one being data pre-processing. This can improve the performance of the network and in some cases remove the need for a non-linear solution entirely.

4.4.7.1 Data Preprocessing

The efficiency of an ANN, during both training and operation, can be improved by first pre-processing the data before it is fed into the network. By doing this, the ANN can be made to focus on the desired qualities of the data and not waste time on irrelevant features that could be removed prior the ANN stage. Some techniques can assist the network by concentrating the data into the active region of the network. These include some of the following:

- *Data Normalisation.* This essentially involves normalising the data so as to place it within the active region of the neurons activation functions.
- *Time Domain Filtering.* This can be used to remove random noise or "outliers (spikes)" from the data. This focuses the ANN on the frequency range of interest
- *Functional Links.* This are generally additional inputs to the ANN and consist of functions (typically nonlinear) of inputs (e.g. trig functions or polynomial nonlinearities). This saves the network time as it doesn't have to generate these terms itself.

Other more complicated techniques can also be employed such as differencing / deseasonalisation of data and linear transformation/complexity reduction techniques. The aim of these techniques is to remove unimportant features from the data thus allowing the ANN to concentrate on the desired information.

Differencing/Deseasonalisation of data

This approach is generally used in the case of dynamic (time series) data. This was inspired by the Box-Jenkins [60] methodology for time series model building. By applying some of the operations below repeatedly, the data can be made stationary, thus making it easier to analyse. There are several operations that can be applied.

- The first of these is to remove the d.c. component from the data. This is done by subtracting the average value from all the data points. This gives a smaller dynamical range and allows the ANN to focus on the variations in the data.
- Trend removal can also be applied to the data. This involves fitting a first order polynomial to the data and then subtracting it from the data.
- Differencing of the data, which involves subtracting each value from the following one, is equivalent to discrete-time differentiation. The differencing operator is given by:

  $$\Delta^d(z) = (1 - z^-1)^d \tag{4.30}$$

  where d is the order of the differencing operator. Differencing helps to remove non-stationarities from the data.
- Deseasonalisation of the data involves subtracting each value from values a "season" ahead. This can apply to time-series data that has a periodic element to it. The deseasonalisation operator is given by:

  $$\Delta_L^D(z) = (1 - z^-L)^D \tag{4.31}$$

  where D is the order of the deseasonalisation operator and L is the season length.

While these methods help to remove non-stationarities for the data, the differencing and deseasonalisation operations reduce the data set by one and a season respectively. Thus it is important to being with a large data set.

Linear Transformation/Complexity Reduction

A key problem in statistical pattern recognition is that of feature selection or feature extraction. Feature selection refers to a process whereby the data space is transformed into a feature space that, in theory, has exactly the same dimension as the original data space. However, the transformation is designed in such a way that the data set may be represented by a reduced number of "effective" features and yet retain most of the intrinsic information content of the data; in other words the data set undergoes a dimensionality reduction [7].

Principal Components Analysis (PCA) (also known as the Karthounen-Louve transformation) can be used to achieve this. PCA makes use of a mathematical tool called Singular Value Decomposition (SVD). SVD decomposes a matrix, G as given by:

$$G = U\Sigma V^T \tag{4.32}$$

where $U \in R^{NxN}$ and $V \in R^{MxM}$ are orthogonal singular vectors and $\Sigma \in R^{NxM}$ is a matrix of singular values.

In terms of the data, the SVD transforms the data onto an orthonormal basis representing the main directions in the data. Now strong features in the data can be organised before applying them to the network. One technique is data rotation. The data in P (the input matrix) is rotated. The elements of each column in the new P matrix (i.e. the rotated P matrix) can be interpreted as the amount of influence or gain of these main directions are present in X.

When the transformation has been completed, a reduced set of significant inputs can be determined by examining the singular value spectrum.

Data preprocessing is a very important part of ANN design. By manipulating the data before presenting it to an ANN, the size and complexity of the network can be significantly reduced.

4.4.7.2 Choosing a network type

When the decision has been made to apply a neural network to a problem, the next step is to decide upon a network topology or set of topologies. Table 4.1, shows three problems types which ANNs are suitable for displays which topology is suited to each.

Network Type:	Function Approximation	Classification	Associative Memory
MLP	✓	✓	-
RBF	✓	✓	-
SOFM	-	✓	-
Hopfield	-	-	✓

Table 4.1: Table showing network suitability to three main problem types

Even when classified like this it is still not obvious what topology to use. This final decision may be based on trial results of several architecture or indeed on personal experience.

4.5 ANNet Course

4.5.1 Course Layout

The ANN course is broken up into chapters with each one describing a different ANN architecture. They are as follows:

- Introduction to ANNs
- Simple neural networks
- Multi-Layer Perceptron networks

- Radial Basis Function networks
- Dynamic neural networks
- Self-organising Feature Maps
- Hopfield networks
- Data pre-processing
- Application of ANNs

The chapters are presented in a logical progression. The course begins with a general introduction to the field of ANNs, including some background and applications. The next topic introduced is that of simple neural networks. These provide the basis for several of the proceeding network types. The chapter also introduces some foundation learning algorithms. The next logical progression is to that of multi-layer networks. This chapter builds on previous material and explains the suitability of these ANNs to both function approximation and classification problems. Radial basis function networks are then covered, as they too are suited to similar problem types. Dynamic neural networks (MLP and RBF) are extensions to the above networks in that they allow for dynamic data to be handled.

Finally two other network topologies which are introduced. These are the Hopfield network, as an example of an associative memory ANN and the self-organising feature map network, as an example of feature extraction and vector coding ANNs. The course finishes with a more general chapter on data pre-processing and a brief guide to the application of ANNs. These can be applied across all network types and so is presented in a general fashion. Appendix A shows a complete breakdown of the course in to its chapters, sections and subsections.

4.5.2 Characteristics of ANN Course

The subject of ANNs has many characteristics that lends itself to a computer-aided learning environment. Many of its concepts can be illustrated via interactive examples, whereby the user can see, in a visual way, the effects of parameter changes on the network performance. By using the Internet as a platform, Java in the form of Java applets can be used for the development of such examples. This is well illustrated in the Hopfield network Java applet, (see Figure 4.13). The demonstrations convey the concepts behind each of the ANNs in a hand-on and visual fashion. This opinion is shared by Ryan Danell [61] by his request to use the Hopfield demonstraion in one of his presentations.

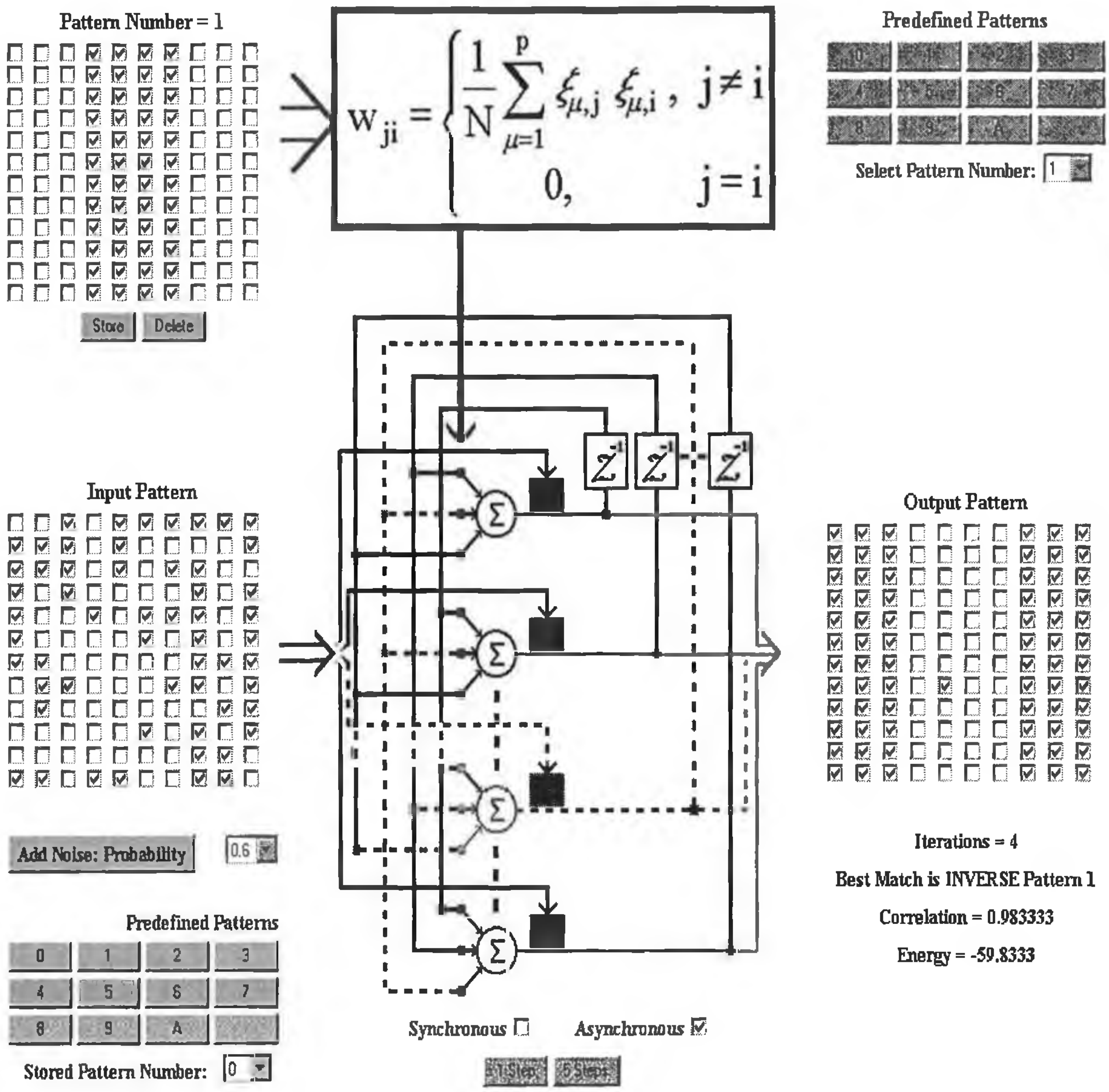


Figure 4.13: Hopfield Java Applet

In this associative memory example, the user, having stored a number of predefined or user-defined patterns, can examine the recall abilities of the network by initialising the network with noisy versions of the stored patterns. The network then iterates to a stable state, which could be the required stored pattern or an alternative stored pattern, an inverse pattern or a spurious pattern. This particular example is very visual and gives the user a good insight into the operation of the network.

As seen with all the ANN topologies, there is a significant amount of computation involved in their training and operation. The computation engine of MATLAB™ [63] (described later), provides an ideal environment upon which to design and develop ANN examples. Examples similar to the Java applets, can also be developed in MATLAB™ [63] providing the user with graphical interpretations of the ANN operation.

Many concepts and models appear repeatedly throughout the ANN course. These include various learning algorithms and neuron models. An ideal method of interlinking such ideas in a non-linear fashion is the hypertext approach as described in Section 2.2.2. Via this network of interlinked documents, the user can, by following hyperlinks, refer back and forth to relevant, related topics.

4.5.3 General Requirements

The general requirements of the ANN course can be laid out as follows:

- **Hypertext:** The nonlinear approach provided by hypertext is an important element within the ANN course. As described in Section 4.5.2, the ANN course has many topics which are relevant to several different areas. Through the use of hypertext, these topics can be interlinked, allowing the student to refer back and forth where necessary.

- **Online Quizzes:** As outlined in Section 2.2.2.1, online quizzes are very beneficial to students. However, as ANN theory is very mathematical, considerable effort was required during the generation of the these quizzes. While the more factual type questions proved relatively straightforward, the mathematical ones required more thought. The questions had to be designed in such a way as to give benefit to the student without over burdening them with large amounts of computation. Suitable and plausible alternative answers also had to be generated to force students to think through the problem and avoid *guessing* the answer.

- **Interactive Examples:** Many of the concepts and ideas within ANNs can be best understood through practical examples. By changing the parameters of the networks and seeing in graphical form the consequences of these changes the student obtains a clearer understanding of these concepts. The mathematics and theorems prove these ideas to be true, but the practical examples can often help the students to grasp the ideas and cement them in their minds. Thus the Java applets and MATLAB™ [63] examples are essential requirements of the ANN course as they give the students the opportunity to implement the ideas that they have been studying.

- **Communication:** The ANN course is taught both as part of the RACeE programme and also as a module in the M.Eng Masters Programme at DCU. For remote students studying via RACeE, it is important that they be able to communicate with others students, as discussed in Section 2.3.2. Through communication the aim is to promote traditional classroom interactions between students and to try and reduce the feeling of remoteness and isolation that some student may feel. A common RACeE/M.Eng mailing list encourages collaboration between both sets of students and the weekly summaries of in-class progress act as a pacing guide for the distance students.

Chapter 5

The CAL Environment

5.1 Introduction

In order to develop the CAL package described in Section 3.4, it is necessary to examine some of the tools and issues involved. The Internet, despite is many advantages, still poses some problems for the online course designer. This chapter will examine these issues as well as providing a comprehensive background into the tools used in the development of the ANNet course and the motivation for their choice.

This chapter also presents a detailed examination of the Internet and the WWW. It outline the use of Perl in the development of interactive quizzes and describes the MATLAB™ engine and its role within the CAL package.

5.1.1 Environment Requirements

As discussed in Section 2.2.3, in order for a CAL environment to be effective it should incorporate multimedia features into an interactive, hyperlinked framework. Viewing these features should require minimum effort on the part of the student, as should the navigation through the course material.

Access to the course material should be widespread and independent of computer platform. Also, the development of material on the part of the educator should not infringe upon or be inhibited by the nature of the environment. In addition, the educator should be able to develop material in an incremental fashion and where suitable be able to reuse existing material in other courses.

As shown in Chapter 2, the Internet provides a platform that meets the above requirements and was thus chosen for this project.

5.2 The Internet

5.2.1 A Brief History

The Internet was originally developed by the US Department of Defence (DoD) in the 1960's as a means of communicating over a reliable and decentralised network that linked the DoD with military contractors and universities conducting military research. Known as ARPAnet (Advanced Research Projects Agency, part of the DoD), it was a text based communication network providing file transfer and electronic mail capabilities [20]. For the most part it was widely unknown, with only 4 host computers in 1969. During the 1970's, it became available to non-military organisations such as universities. Its growth continued until in the late 1970s the network standard had to be changed to meet demand. The original protocol, called Package Switch Nodes, could not support such growth and was replaced by the TCP/IP (Transmission Control Protocol/Internet Protocol) communications standard [64].

In the late 1980's several US and foreign networks were becoming interconnected, thus forming what is now known as the Internet. The same TCP/IP protocol was used. In 1990 the ARPAnet was shut down and the administration of the Internet was taken over by NSFNET (US National Science Foundation). In 1991 the ban on commercial use was lifted and by 1994 commercial networks dominated the global Internet scenario [20]. A survey carried out [65] shows that, as of July 1998, there were nearly 36.74 million Internet hosts worldwide.

5.2.2 The Internet Platform

The Internet is made up of four important services [66], notably

- *mail*: The mail service allows for the transmitting of messages
- *telnet*: Allow for the establishment of a terminal session with a remote computer
- *FTP*: FTP (File Transfer Protocol) allows for the transfer of files between computers (generally remote computers)
- *general client/server facility*: A *client* program can request help or the use of *server* programs, with both client and server computers being remote from each other.

Based on these services the Internet can provides many other resources such as Electronic Mail, Remote Login, Usenet, FTP, Gopher, etc.

The Internet platform is independent, which means that information on the Internet can be accessed by a client machine regardless of the operating system of that machine. This is achieved through the TCP/IP protocol.

5.2.3 The World Wide Web

One of the most important features of the Internet is the World Wide Web (WWW). The WWW began in March 1989, when Tim Berners-Lee of CERN [67](a collective of European high-energy physics researchers) proposed the system as a means of transporting research and ideas effectively throughout the organization [68]. The growth of the WWW into what we know today began with the development of the Mosaic *web browser* at the University of Illinois National Centre for Supercomputing Applications (NCSA) in 1993. In the first 6 months over 2 million copies of the browser were downloaded [4]. Since then the growth of the WWW has been phenomenal.

The WWW accesses HyperText Mark-up Language (HTML) documents via an address in the form of a URL (Universal Resource Locator). This takes the form of: http://domain-name/directory/sub-directory/file.html. The URL for the ANNet course, for example, is given by http://www.eeng.dcu.ie/~annet/index.html. Unlike the text only capabilities that existed before, the WWW is capable of housing images, sound and video within its HTML pages. Its hypertext foundation allows *web site designers* to interlink related documents via *hyperlinks*.

An important distinction is that the WWW is not the Internet, rather a graphical and navigational tool that exists on top of the Internet. The web browser interprets the HTML files and displays the information in the form laid out within the HTML code.

5.2.4 HTML and HTML 4.0

HTML is the lingua franca for publishing hypertext on the World Wide Web. It is a non-proprietary format, based upon SGML (Standard Generalized Mark-up Language) and can be created and processed in a wide range of tools from simple plain text editors to sophisticated *wysiwyg* (What You See Is What You Get) authoring tools [68].

HTML uses *tags* to format the HTML documents, with $< html > \ldots < /html >$ being its defining tags. Web browsers can, however, still read HTML documents that do not have these tags. A defined set of these HTML tags are used to format and structure text within the document into headings, paragraphs, lists, hypertext links, etc. Tags also govern the inclusion of images, sound and video clips and tables within a document.

A typical document structure may appear as:

< html >
< title > ... < /title >
< head > ... < /head >

< body >

...

...

< /body >
< /html >

HTML 4.0 is the latest standard of HTML to be recommended by the World Wide Web Consortium (W3C) [68]. HTML 4.0 extends HTML with mechanisms for style sheets, scripting, frames, embedding objects, improved support for right to left and mixed direction text, richer tables, and enhancements to forms, offering improved accessibility for people with disabilities [69]. Of particular interest to science/engineering disciplines is its inclusion of the Greek symbol set. This allows for the inclusions of such symbols within online publications.

Once adopted by the major commercial web browser companies HTML 4.0 will offer more freedom and flexibility to web publishers

5.2.5 Server Side Includes

Server Side Includes (SSI) are features that are added to a standard HTML document by the server before it is sent to the client, i.e. the web browser. The server parses the document for SSI tags and when it finds one it replaces it with the information that it represents. These tags can represent CGI scripts (discussed in Section 5.4.4), other HTML files or even *environment variables.* These environment variables may contain data such as the last time the document was modified, the date and time or even data about where the document was accessed from. HTML files containing data that is common to many files can be written to one file and via an SSI be included in all the documents, thus saving space on the server.

Each SSI has the following format:

<! – –#command tag1="value1" tag2="value2" – – >

Each command takes different arguments and most only accept one tag at a time. For example to include another HTML document within your current document the command line might read:

<! – –#include file="footer.shtml" – – >

All SSI documents have the extension ".shtml" as opposed to the standard ".html" one. This informs the server that the document may contain SSI tags and so should be parsed for them.

5.2.6 Java and Java Applets

Java is a high level programming language developed at Sun Microsystems in the early 1990's. It is a 32-bit programming language that was originally developed for use in the consumer electronics market. Existing object-oriented languages such as C++ were deemed unsuitable due to their inherent historical flaws and complexity. Their solution was to design a simple, more refined language that was not hindered by compatibility yet still remained familiar. That solution became Java [70].

The Java language and its environment are best understood by the attributes they embody: familiar, simple, object-oriented, architecture-neutral, portable, interpreted, high performance, threaded, robust, secure and strategic [70].

Java programmes can run as stand alone applications, just like any other programming language, however it is Java's influence on the WWW that is of particular interest in this project. Java programmes that run over the WWW are called *Java Applets.* These are compiled executable Java programmes that can be embedded in HTML documents using the HTML applet tag given by

< APPLET CODE="filename.class" HEIGHT=300 WIDTH=300 >< /APPLET >

When the HTML page is requested by the web browser, this Java *class file* is also downloaded and runs from within the HTML document. These Java applets can contain anything from coded animations to computational engines and plotting facilities as shown in [1].

5.3 Equations and Mathematical Symbols

The concepts and theories that define science and engineering based subjects are inherently mathematical in their descriptions. The ability to include such mathematical equations and symbols within any course, within these disciplines, is essential. The subject of ANNs in no different in this respect. In order to properly describe the working of these networks in a concise and clear manner, mathematical descriptions have to used.

Although the Internet was originally developed to aid scientific research at CERN [67], it never catered for the handling of mathematics. Only recently is this problem beginning to be addressed. Some of the ongoing and future methods of overcoming this problem are discussed below.

5.3.1 Image Files

The most common approach to incorporating mathematics in to web documents has been via image files. The equations would be created in one package and saved, usually a ".gif" (Graphics Interchange Format) file, for inclusion in the web document. The HTML tag for the inclusion of such an image would appear as

< IMG SRC="image.gif" HEIGHT=20 WIDTH=20 >

This is not a very efficient approach for several reasons. Firstly, image files take up considerably more space on the web server than the text based HTML documents. If several images were to be included in the one document, as is quite reasonable in heavily mathematical subjects, the download time for the page would be significantly increased. This becomes more evident to users who are accessing the document remotely over a low speed modem. An example of such an image can be seen in Figure 5.1.

$$y = \frac{2}{1+e^{-\beta z}} - 1$$

Figure 5.1: Image file of mathematical equation

Another problem associated with including graphics in this format is the lack of control the document designer has over the font type of the equations. Whereas the font and size of text can be varied from within the browser, the size of the image cannot. Thus the document may not always be viewed in the exact font type and size that the designer had intended and so equation may appear disproportionate from the text. As can also be seen from Figure 5.1 the equation appears grained. This is due to the fact that the equation was originally captured in bitmap format.

Despite these drawbacks this approach was adopted by the author due to the lack of efficient and commercially acceptable alternatives. New techniques and technologies such as HotEqn and MathML offer future solutions to this problem.

5.3.2 HotEqn

HotEqn is a Java applet that caters for the inclusion of mathematical equations and symbols within a HTML document. The software makes use of the LaTeX[71] notation to code its examples. The applet scans and parses LaTeX[71] strings, which are entered as parameters to the applet. It interprets the string and the relevant fonts are downloaded from the server where the applet is stored. Figure 5.2, shows the equation output from the HotEqn applet

$$y = \frac{2}{1+e^{-\beta z}} - 1$$

Figure 5.2: HotEqn form of mathematical equation

This is generated from the following lines of code which are entered in the HTML document.

< applet code=HotEqn.class height=53 width=105 name=”myeqn” align=”absmiddle” >
< param name=”equation” value=”y=\frac{2}{1+e^{-\beta z}}-1” >
< param name=”Fontname” value=”Helvetica” >
< param name=”Fontsize1” value=”18” >
< param name=”fgcolor” value=”000099” >
< param name=”bgcolor” value=”FFFFFF” >
< param name=”border” value=”FFFFFF” >
< param name=”halign” value=”center” >
< param name=”valign” value=”middle” >
< */applet* >

HotEqn allows the size and font of the equation or symbol to be set as well as the fore and background colours.

Even though HotEqn does not use images, it has to interpret the LaTeX code within the HTML document and then download the relevant fonts. This puts extra load on the web server as it has to first deliver the document that was requested and then having done that deliver the fonts requested by HotEqn. For this reason HotEqn was not used in place of the standard image files.

5.3.3 MathML

In early April, 1998, the World Wide Web Consortium (W3C) [68] recommended the Mathematical Mark-up Language (MathML) [73] as the new official standard for delivering mathematical content over the WWW. A W3C recommendation indicates that the specification is stable, contributes to the WWW interoperability, and has been reviewed by the W3C membership, who are in favour of its adoption by industry [72].

According to the W3C [68], the aim of MathML is not just provide control over the visual presentation of mathematical notation. It also aims to provide a set of controls to convey the intended meaning or content of the notation. The W3C write:

> "Mathematical ideas exist independently of the notation that represents them. The challenge in putting math on the WWW is to capture both notation and content in such a way that documents can utilize the highly-evolved notational practices of print, and the potential for interconnectivity in electronic media" [73].

For instance, consider the equation in Figure 5.2. The MathML code for this would be generated by:

```
< mrow >
< mi > y < /mi >
< mo > = < /mo >
< mfrac >
    < mn > 2 < /mn >
    < mrow >
        < mn > 1 < /mn >
        < mo > + < /mo >
        < mi > e < /mi >
        < msup >
            < mrow >
                < mo > - < /mo >
                < mi > &beta z < /mi >
            < /mrow >
        < /msup >
    < /mrow >
< /mfrac >
< /mrow >
```

This is the *presentation mark-up* of MathML which was generated in the AMAYA [74] browser. Figure 5.3 shows the form in which this MathML equation would appear.

$$y = \frac{2}{1 + e^{-\beta z}}$$

Figure 5.3: MathML form of mathematical equation

Amaya is a developmental browser which is not yet completed and so the intended mark-up was unavailable. This intended meaning within MathML aims to not only make the equations searchable through their meaning, but also to make it compatible with other interfaces such as graphical displays or computer algebra systems [72]. As stated, currently only developmental browsers such as Amaya cater for MathML. When adopted by commercial browser designers, MathML will greatly improve the quality and efficiency of mathematical content on the WWW.

5.4 Programming WWW Applications

5.4.1 Choice of Perl

In order to generate the online tutorial system and also to keep track of student usage of the course, software programs had to be developed that would run on the web server and that could be executed from the web pages. The software also had to suitable for scanning large amounts of data and extracting relevant information and re-formatting this information for future examination. C and Perl are the two most commonly used languages for these applications. Perl, however, was found to have many of the features of C that were required for this particular application as well as its own well known text processing abilities. It does not require compilation and is portable. In order to make C portable several #ifdef clauses are required for different operating systems. Thus Perl was deemed the more suitable of the two for the particular applications required by this project.

5.4.2 Introduction to Perl

> "Perl is designed to make easy jobs easy, without making hard jobs impossible" [75].

Perl is an interpreted language optimized for scanning arbitrary text files, extracting information from those files and printing reports based on that information [76]. It was initially designed as a glue language for the UNIX operating system by Larry Wall in 1986 and is now one of the most portable programming languages available today [75].

Another advantage of Perl is that it is freely available, and freely redistributable. It can be obtained as source code or as precompiled binaries for many operating platforms.

5.4.3 Relevant Features

As mention in Section 5.4.1, the application required by this project is that of interfacing with web pages and then extracting necessary information which is to be further written to and stored in various files. Two of the important features of Perl required by the project, notably filehandling and data manipulation, are described below. Perl interaction with web documents is described in Section 5.4.4

5.4.3.1 Filehandling

In order to write to a file, Perl makes use of filehandles. These are merely names given to a file, device or socket so as to remember which file is in use. Perl allows files to be opened for several purposes, notably, reading, writing and appending. The Perl commands for these are shown below [75]:

open(HANDLE, " >> filename")	Open file and append text to it
open(HANDLE, " > filename")	Open file for writing to

open(HANDLE, "filename")	Open file for reading
open(HANDLE, " < filename")	Open file for reading

The *open* function opens a file. The first parameter is the filehandle which allows Perl to refer to the file in future. The second parameter is an expression denoting the filename. The characters before this filename determine what way the file will be manipulated. In order to write the file the syntax is as follows:

```
$file = '/etc/passwd';     # Name of the file
open(INFO, $file);         # Open the file
print HANDLE " ... ";      # Write to file
close(INFO);               #Close file
```

5.4.3.2 Data Manipulation

Perl is very efficient at manipulating strings of data. It can read in data from a file, or other device, in single lines or as one large string, which can then be broken up into smaller more meaningful strings depending on the type of data and application in question. These features are particularly useful with regard to the ANNet course. As will be described in the following sections, the ability to transfer and format data between files is vital.

5.4.4 Common Gateway Interface

Common Gateway Interface (CGI) is the standard used to run programs for a client (web browser) on the web server. They are often coded using Perl, thus they are sometimes referred to as *CGI scripts* [76]. When evoked, CGI programs return dynamically generated responses depending on the inputs they receive. Standard HTML documents are *static* in the sense that they reside on the web server until they are requested. The content of the document does not changed unless physically altered. CGI offers the web designer the ability to create documents *on the fly* (dynamically) based on certain conditions.

The CGI programs are activated when the user selects its URL. By doing so the user requests the web server for an action. This request is processed by the server by running the program. Once the program has run, it returns its results to the browser that made the request. The most common use of these CGI programs is in processing information received from HTML forms. Through the browser the user enters information in the form and then *submits* (i.e. selects the URL of the CGI program, usually in the form of a HTML button) this to the web server for processing. Having done this, the program returns a response based on the data it has received. These responses are pre-coded within the program but the exact response returned to the browser depends on the input information it received. The format of the response can be plain text, binary data or HTML documents.

CGI programs can also be evoked without action from the user. To do this SSI (see Section 5.2.5) are used. The SSI command

<! – –#exec cmd="cgi-prog.pl" – – >

automatically evokes the CGI program as soon as the document is requested.

This ability to dynamically create documents and process the information received from them is of particular interest within the context of the ANNet course for the development of multiple-choice online quizzes. Predefined questions can be displayed and corrected by the CGI program and the results returned to the user. These programs can also be used to collect usage data from HTML documents, which is again of use for evaluation purposes of the ANNet course. These are described in more detail in the following sections.

5.4.5 Multiple Choice Quiz Generation

In order to generate the online quizzes described earlier in this work, the author made use of Perl CGI programs. For each chapter within the ANNet course, a set of multiple-choice questions were drawn up along with the corresponding answers.

The questions were designed following the guidelines set down in [77]. Where pos-

```
sub question_9
{
print "<FONT SIZE=+1><P>
       The Perceptron learning rule represents a:
       </P><OL>";
print "<LI><INPUT TYPE=\"radio\" NAME=\"q9\" VALUE=\"a\">
       Supervised</LI>
       <LI><INPUT TYPE=\"radio\" NAME=\"q9\" VALUE=\"b\">
       Unsupervised</LI>
       <LI><INPUT TYPE=\"radio\" NAME=\"q9\" VALUE=\"c\">
       Hybrid</LI></OL>
       <P>learning rule.</FONT>";
}
```

Figure 5.4: Format in which questions were stored in Perl script

sible four alternative answers were provided, with each of the alternatives being plausible answers to the question. The position of the correct answer was varied with all the possible answers being consistent and approximately equal in length.

For each chapter within the course, a number of questions were drawn up and entered in the Perl CGI program as individual function or subroutines, as shown in Figure 5.4, with each question being written in HTML format.

Ten questions are chosen randomly from the question set and formatted by the Perl CGI program for printing to the web browser. Each time the quiz is accessed a different set of ten questions is chosen. The quiz is structured as a HTML form. When the students have completed the quiz they submit it for correction. The *"NAME"* tag within each question identifies it within the correction process. The program also adds textfields to the quiz to allow the students to enter their details such as name, email address and student number. An example of such a quiz can be seen in Figure 5.5 (see page 73).

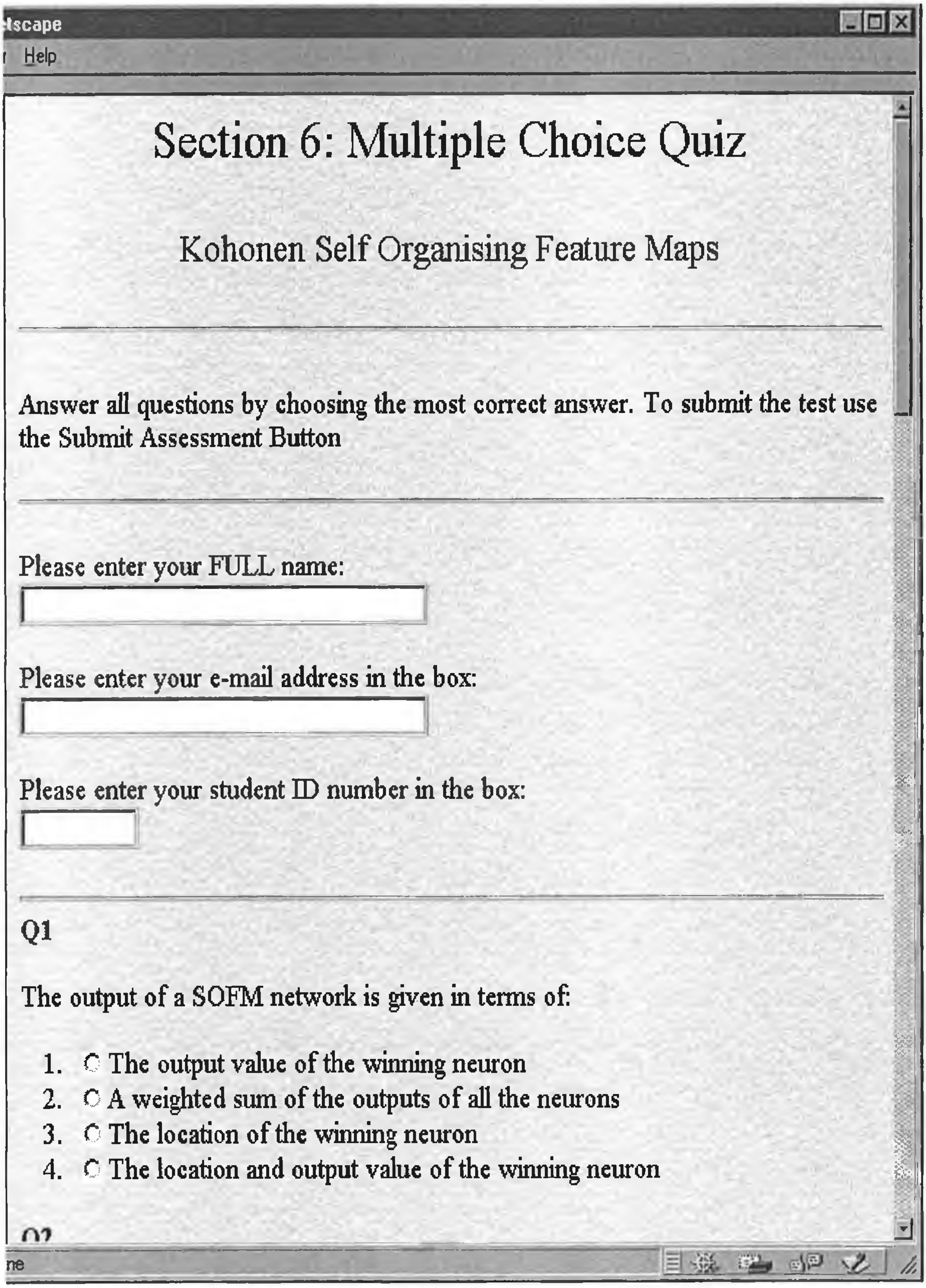


Figure 5.5: Screen shot of multiple choice quiz

5.4.6 Multiple Choice Quiz Correction

Once the multiple choice quizzes have been generated, a system must be set up to correct the quizzes and give the student feedback from them. The flow chart below outlines the procedure adopted for this task.

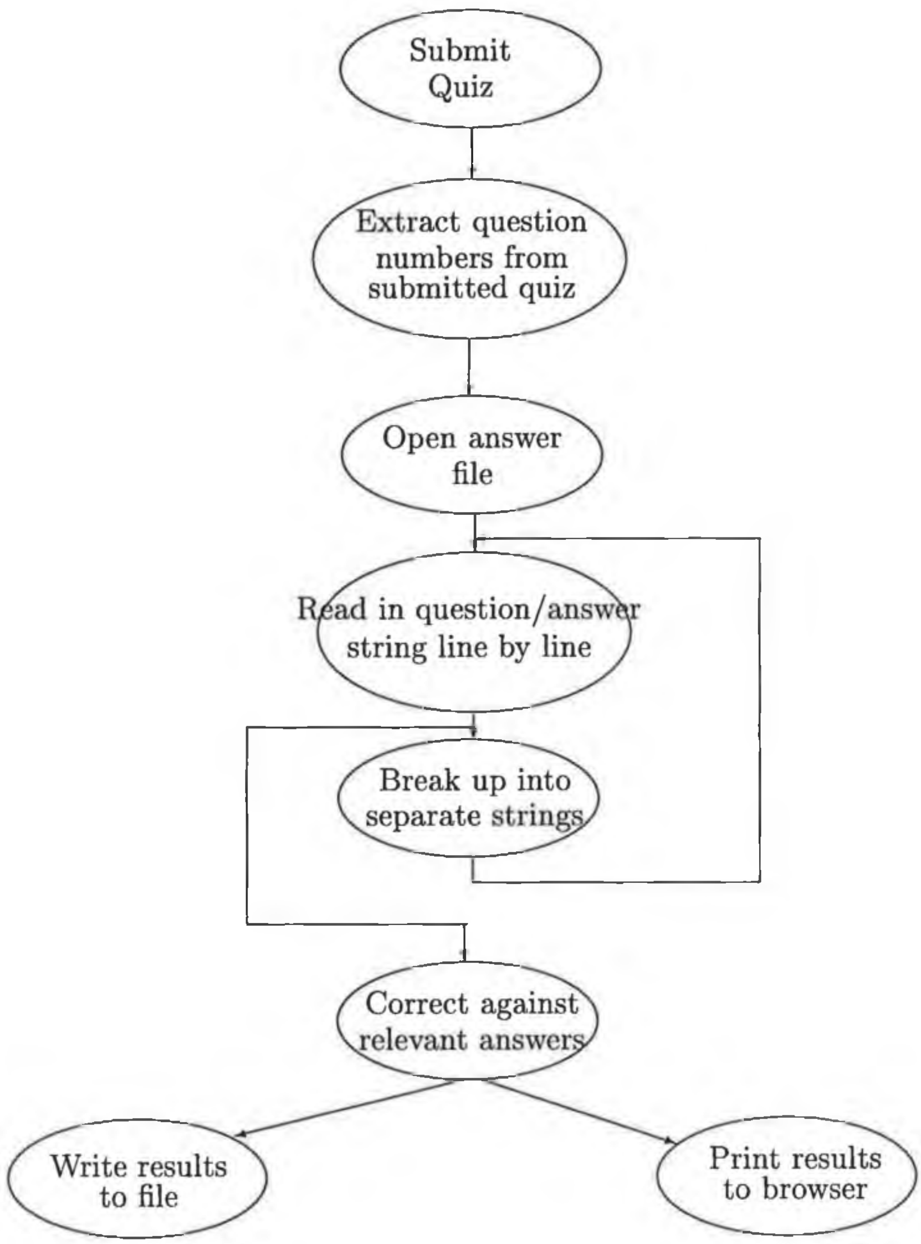


Figure 5.6: Correction program flow chart

As the questions are generated randomly from the set, a handle must be kept on which questions are in use at any one time. To do this the question numbers that are active are printed to the HTML page as hidden form variables

< INPUT TYPE="hidden" NAME="question numbers"
VALUE="2 12 10 7 3 17 23 5 11 19" >

Having read in these values and the users answers from the quiz, the answer file is opened. This is a formatted text file with the question numbers and answers stored in the following way

Question1=d
Question2=c
Question3=c
Question4=a
Question5=c

On opening the file the first task is to separate the answers from the question numbers. This is done using the following Perl command:

```
($name, $ans) = split(/=/, $line);
chop $ans;
@answers[$i] = $ans;
@quest_names[$i] = $name;
```

This splits the string with the = sign as the delimiter. This is done on a line-by-line basis with each answer and question number being stored in the arrays *answers* and *quest_name* respectively. Likewise all the users answers are pushed into an array. The program then uses a *string compare* routine to compare the user answers against the correct answers, incrementing a counter with each correct answer and storing the incorrect answers in an array. This allows the program to inform the user of the questions they answered incorrectly.

The results are then written to a results file for perusal by the lecturer. This is a comma delimited text file of the form:

name, email, student id. num., time and date, answers given, grade

The benefit of this format is that the data can then be imported into a spreadsheet package such as EXCEL for further processing.

Finally, the results are returned to the browser in HTML format to inform the student of the grade. It gives them a mark out of ten and informs them which questions they answered incorrectly. They can then attempt these again or go back and revise the subject area. Figure 5.7 shows a typical response that might be returned to the browser.

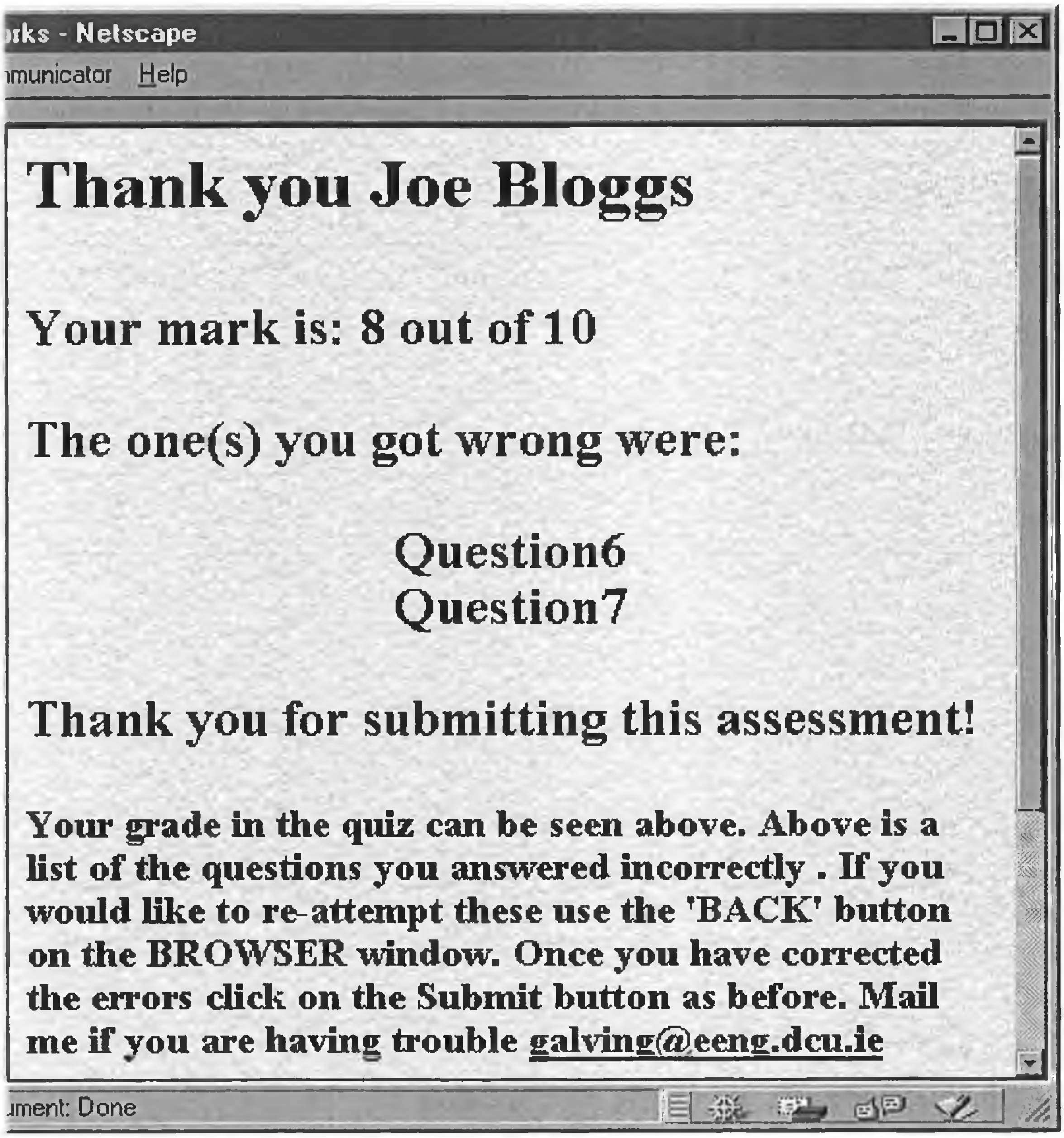


Figure 5.7: Screen shot response form the correction program

5.4.7 Statistics Generation

As described in Section 5.4.4, CGI programs can be run, in conjunction with SSI directives, as soon as a page is requested by the browser. This feature was exploited by the author to compile statistics about the usage of the web course by students. Each time a page was accessed by a student, this action was logged by a Perl CGI script embedded in the page using SSI. The program logged the username (i.e. password username, described in Section 5.5), the page accessed, the date and time, the location accessed from and the type of browser used. These were then written to an access file for evaluation. This, like the results file from the quizzes, was a comma delimited text file of the form:

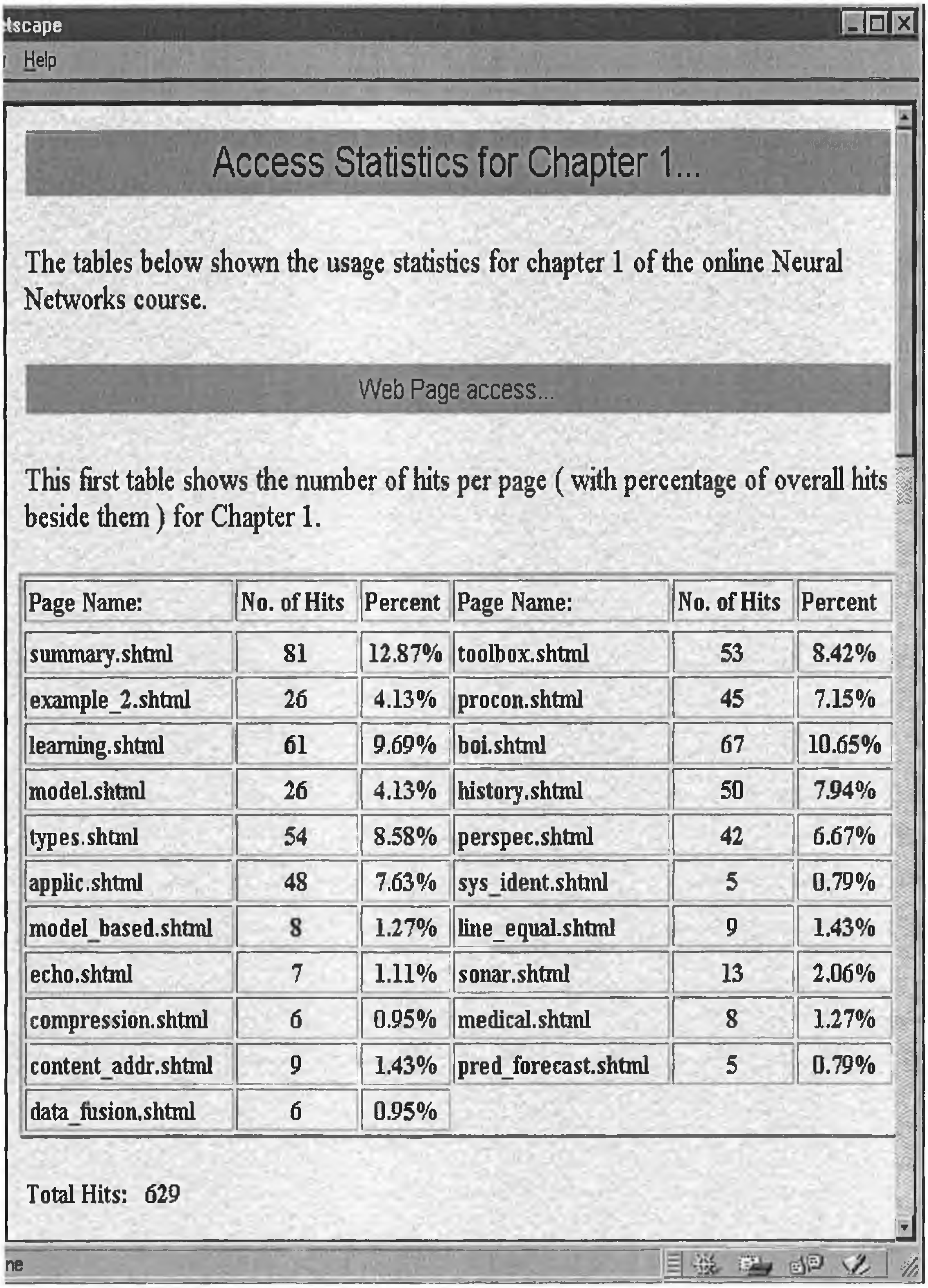
username, page accessed, date and time, remote host, browser type

Information such as remote host or page accessed is obtained through the web servers *environment variables.* These allow the web server administrator to monitor users of its web documents. They can be accessed from within a Perl CGI script. To assign one of these variables to a standard Perl variable an action similar to the following operation is carried out:

```
$page_accessed = $ENV{'SCRIPT_NAME'};
```

The Perl variable $page_accessed can now be saved along with the others (as listed above).

As well as saving the statistics to file, a page was setup at the start of each chapter which displayed that chapters statistics in tabular format. The tables contain sets of information, notably the number of hits per page, the browser type used to access the pages, the remote hosts from where the pages were accessed and finally the number of hits per user. An example of this can be seen in Figure 5.8 (see page 78).



tscape

Help

Access Statistics for Chapter 1...

The tables below shown the usage statistics for chapter 1 of the online Neural Networks course.

Web Page access...

This first table shows the number of hits per page (with percentage of overall hits beside them) for Chapter 1.

Page Name:	No. of Hits	Percent	Page Name:	No. of Hits	Percent
summary.shtml	81	12.87%	toolbox.shtml	53	8.42%
example_2.shtml	26	4.13%	procon.shtml	45	7.15%
learning.shtml	61	9.69%	boi.shtml	67	10.65%
model.shtml	26	4.13%	history.shtml	50	7.94%
types.shtml	54	8.58%	perspec.shtml	42	6.67%
applic.shtml	48	7.63%	sys_ident.shtml	5	0.79%
model_based.shtml	8	1.27%	line_equal.shtml	9	1.43%
echo.shtml	7	1.11%	sonar.shtml	13	2.06%
compression.shtml	6	0.95%	medical.shtml	8	1.27%
content_addr.shtml	9	1.43%	pred_forecast.shtml	5	0.79%
data_fusion.shtml	6	0.95%			

Total Hits: 629

ne

Figure 5.8: Screen shot of online usage statistics

5.5 Password Protection

The MEng and RACeE programmes, like all courses offered by DCU are subject to fees. As the Internet is available to anyone with a modem, PC and Internet service provider, some form of restriction had to be imposed on the ANN web site. Only fee paying students were to allowed access to the course. To achieve this, the password protection facility of the web server was used. This involves setting up two files, .htaccess and .htpasswd. The .htpasswd file is where the user passwords are stored. The .htaccess file provides information about where this file is located an also contains the names of the users who are allowed access to the site as shown below.

AuthUserFile /user/mailing/annet/conf/.htpasswd
AuthGroupFile /dev/null
AuthName Artificial Neural Networks - Course EE551
AuthType Basic

require user usename1 username2 ...

The position of the .htaccess file within the web site directory structure dictates what files are to be protected. For example, in the directory tree in Figure 5.9, all files and directories below the public_html directory would be password protected.

```
annet
 |
 └──→ course
        |
        └─── public_html
             .htaccess |
                       └──→ Chapter 1
                            Chapter 2
                            Chapter 3
```

Figure 5.9: Example directory tree with password protection

To add a user to the file the **htpasswd** command is used. It adds the username to the .htaccess file and the corresponding password to the .htpasswd file, for example:

htpasswd /user/mailing/annet/conf/.htpasswd -c username

Having entered this command line, the program asks for a password to be entered.

Once password protected, anyone who tries to enter the site will be presented with the dialog box shown in Figure 5.10. Those entering the correct username and password are able to proceed into the course while those without are refused access.

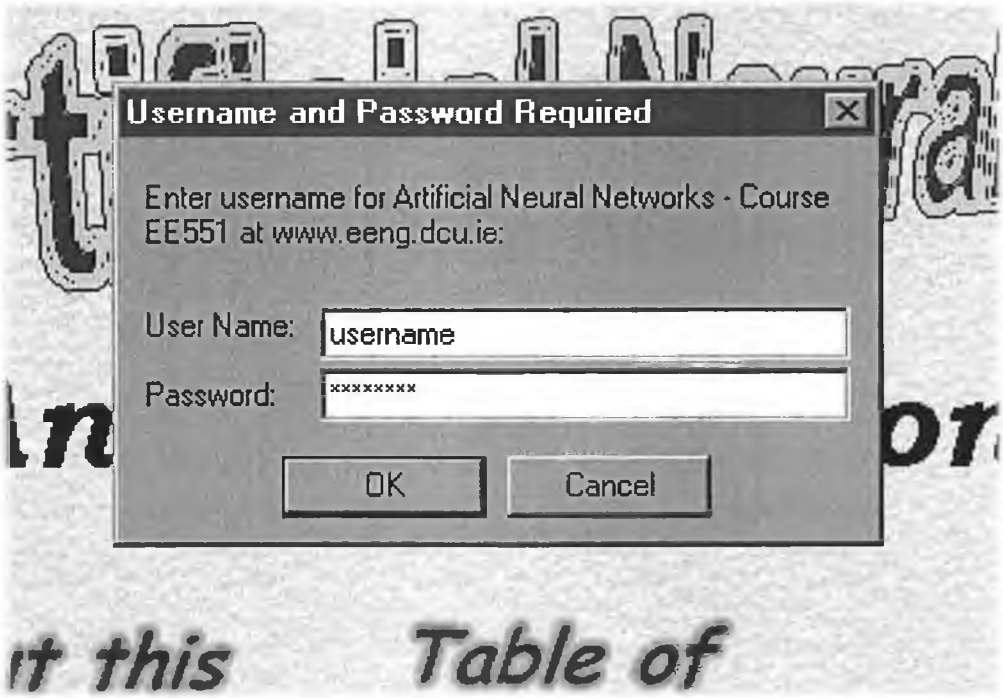


Figure 5.10: Password dialog box

5.6 Archived Mailing List

The archived mailing list facility within the ANNet course was set up using Hypermail [62]. Hypermail is a program that takes a file of mail messages in UNIX mailbox format and generates a set of cross-referenced HTML documents. Each file that is created represents a separate message in the mail archive and contains links to other articles, so that the entire archive can be browsed in a number of ways by following links [62], (see Figure 5.11, see page 81, for an example).

Figure 5.11: Screen shot of ANNet archived mailing list

The HyperMail facility at DCU is maintained by the web server administrator who set up the mailing list for the ANNet course. The author was then given administrative control over the list. This involved approving all subscribers to the list and also the monitoring and removal of improper submissions to the list.

5.7 The Computational Engine

The requirement for interactive examples as laid down in section 4.5.3, requires a computational engine to run the examples and generate the graphical output. In the case of the Java demonstrations, Java itself provides the computation power. However, as mentioned, MATLAB™ [63] also provides a development platform for such purposes.

5.7.1 MATLAB™

MATLAB™ is a high-performance language for technical computing. It integrates computation, visualisation, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation [78]. MATLAB™ makes use of a set of application specific solutions called *toolboxes.* The toolboxes contain sets of MATLAB™ functions (M-files) relating to particular specialised technologies. These toolboxes cover a wide range of subject areas including control systems, neural networks, signal processing and many more. MATLAB™ also allows users to generate their own functions to suit very specific needs. This feature makes it ideal for coding examples and structuring them in a way that the lecturer deems most beneficial.

MATLAB™ can work in conjunction with a simulation package called Simulink™. This simulation environment allows users to generate models of systems in block format, providing many commonly used elements (i.e. linear, nonlinear, discrete-time and input/output functions) for this purpose. The output of the models developed can be viewed graphically through MATLABs™ highly functional graphing capabilities.

There is also a Student edition of MATLAB™ [79]. This is a reduced size version of the professional version. It contains full support for all the language, graphics, external interfacing and other functionality of the professional version. However, the maximum matrix size is reduced to 16,384 elements [79]. It is aimed at students who wish to avail of MATLAB™ at home on their personal PC's.

5.7.2 Ease of Programming

Files that contain MATLAB™ language code are called *M-files*. M-files can be functions that accept arguments and produce output, or they can be scripts that execute a series of MATLAB™ statements [78].

Writing M-files in either format gives the user great freedom over how to design certain tasks or examples. M-files can be set-up to prompt users for input information using the *input* command. This gives the designer of the M-file the ability to let users alter certain parameters within the file, thus making it interactive. Users can try different values and so examine the limits of operation of particular examples.

MATLAB™ provides comprehensive, built-in graphing capabilities. There is extensive in-line help for each, making their use relatively easy. MATLAB™ also uses C-like control flow statements such as *if*, *while* and *for*. These are ideal for programming the iterative algorithms that characterise much of ANN theory.

5.7.3 Downloadable M-files

Unlike Java, MATLAB™ does not run on the Internet. However, work in this area is ongoing through projects such as VCLab [3]. The student edition of MATLAB™ offers students studying remotely the ability to avail of MATLABs™ functionality, similar to traditional classroom students. Lecturers can place M-files on the Internet for these students to download and run on their home PC's. They can be designed in such a way as to be self-contained, while still maintaining the interactivity described above.

Chapter 6

Course Evaluation

6.1 Introduction

This chapter is concerned with the evaluation of the online ANNet neural networks course. This course was run for both on-campus and remote students over the period February 1st to April 28th, 1998. There were ten on-campus students and one remote student undertaking the course during this period. Throughout its duration usage statistics were collected and at the end a review form was circulated. The evaluation is thus based on the these two sets of data and is followed by a cost-benefit analysis in Chapter 7.

6.2 Course Review Forms

6.2.1 Introduction

Course review forms provide a means of obtaining feedback from students regarding the quality of the tuition that they are receiving. Similar to marketing questionnaires, they are used to illicit information from students in a structured manner, which can be then processed and analysed with a view to improving the overall learning experience.

6.2.2 Course Review Format

The author sought to use a semi-structured review form as described in [80]. This involved a mixture of both questions with predefined answers as well as those where the respondent was free to say what they wanted. The aim of the former question type was to determine specific attitudes to the various components and features within the ANN course while the latter sought to get more general feedback in terms of what they would like to see added/removed from the course, general opinion of the course, etc.

A five-point scale was used as a guide to quantify students responses with regard to many of the questions, with 1 representing negative response or dislike and 5 representing the opposite end of the spectrum, as illustrated in the following example.

> How would you rate your ability to
> navigate your way through the online course ?   1   2   3   4   5

The content of the form dealt with several areas notably, presentation and access to the course, navigation within the course, and the individual course components themselves. Finally, there was a discursive section where the students were asked their opinions and recommendations regarding the course. The actual course review form can be seen in Appendix B.

6.2.3 Course Review Form Analysis

The course review form was distributed to the students at the end of the course. It was also given to one student who, while not undertaking the course itself, used it as a means of learning about ANNs for use in a final year project. Thus, there were eleven respondents to the form.

As described in Section 6.2.2, the review form dealt with several different topics and so shall be evaluated as based on these.

6.2.3.1 Presentation and Access

Students were asked to grade the course in terms of its presentation and access qualities using a five point scale. Table 6.1 shows their quantitative response to this topic.

How would you rate the presentation of the online course with regard to ease of learning?	4.27
How would you rate the online course in terms of access (i.e. delays in downloading pages for viewing)?	4.45

Table 6.1: Quantitative results regarding presentation and access

Although the statistical set is quite small, the results are quite favourable. Access to the course with regard to download time, gained a positive response in particular from the single remote student who rated it with a 5. In terms of presentation, the course was also well received.

6.2.3.2 Navigation within the course

An important feature of an online course is that it be easy to navigate. Poorly laid out courses tend to frustrate students as discussed in Section 2.2.2. Table 6.2 shows the quantitative responses with regard to the ANNet course.

How would you rate your ability to navigate your way through the online course?	4.36

Table 6.2: Quantitative results regarding course navigation

Again bearing in mind the small statistical set, this is a positive response, showing that students were in favour of the chapter/section index that was available to them.

No matter where in the course they were, they were always no more than one mouse click away from the main chapter index.

6.2.3.3 Individual course components

To make the ANNet course more self-contained and functional for students using it remotely, several features were added. These included an archived mailing list (to which all students were asked to subscribe), online quizzes at the end of each chapter, Java demonstrations of various concepts within the course, downloadable MATLAB™ files (also to illustrate concepts within the ANN theory), as well as additional material such as extra links to explain the more complex ideas, downloadable notes in postscript format and weekly summaries to help remote students keep in touch with their on-campus colleague's progress. Table 6.3 shows their quantitative responses to these features.

Benefit gained from MATLAB™ examples	4.09
Benefit gained from Java demonstrations	1.82
Benefit of archived mailing list	3.60
Benefit of downloadable notes	4.30
Benefit of addition links and explanations	3.82
Benefit of weekly summaries	4.00
Benefit of online quizzes	3.45

Table 6.3: Quantitative results regarding course features

As can be seen from these results, the addition of many of the features was well received. In particular, the downloadable notes were considered of benefit to both on-campus and remote students. The on-campus students liked being able to have the notes before class while the remote student commented that they were essential for studying in conjunction with the additional explanations provided. Also of benefit were the weekly summaries, added because the remote student requested some guidelines as to how to pace himself.

Also, well received were the downloadable MATLAB™ examples. This partly due to the fact that their assignments were also MATLAB™ based, thus giving them the opportunity to familiarise themselves with the package, but also for the insight it gave them into the concepts they were studying.

The least well received of the additional features were the Java demonstrations. Many of the students never used them, however, those that did found them beneficial, with a quantitative rating of 3.25.

6.2.3.4 Discursive questions and comments

These type of questions on the review form sought to find out what the students generally thought about the course and gave them the opportunity to suggest changes or additions to it. Although not widely completed by the students, those that did tended to use them to back up the feelings expressed in the previous questions. The main area students wanted expanding upon was in the inclusion of more real-world applications and implementations of ANNs.

6.3 Web Site Statistics

6.3.1 Introduction

As described in Section 5.4.7, a log of all student access to the ANNet course was maintained. A sample of this data can be seen in Appendix C. The purpose of this exercise was to monitor student usage of the ANNet course material with a view to:

- Determining which areas of the course students focus most on. This will give an indication of areas where students may be experiencing difficulties.
- Monitoring students usage of the facilities provided, in particular the online multiple-choice quizzes.
- Evaluating the course based on the statistics obtained through this process.

Over the duration of the course there were *201* logins. From these 201 logins, there were *1982* distinct files accesses. A break down of the individual chapter accesses is shown in table 6.4

File Access Breakdown	
Chapter No.	No. of Hits
1	581
2	401
3	313
4	286
5	70
6	143
7	32
8	35
9	18

Table 6.4: File accesses for each chapter within the ANNet course

6.3.2 Additional Content

Much of the *additional* content material was focused on the first four chapters with the remaining chapters appearing (except for *some* additional material) as in the classroom postscript notes. This is borne out in the access figures, with a large proportion focusing on these first four chapters. This backs up the responses from the course review forms where the students regarded these additional notes as being of benefit (i.e. a quantitative grade of 3.82 from a scale of 1 to 5).

There was a increase in usage for Chapter 6 of the course, however this may be attributed to the fact that the third assignment was based on this chapter. Chapters 3 and 4 also had assignments associated with them thus influencing their usage too.

6.3.3 Accessibility and Cross-Platform Compatibility

As stated in Section 5.1.1, accessibility and cross-platform operation are important requirements for online course delivery. This fact was taken into consideration during the development of the course material. Data logged over the duration of the course shows that several platforms and browser types were used to access the course. This can be seen in Table 6.5

Browser and Platform Access	
Browser and Platform Type	No. of File Accesses
Netscape/4.0 (compatible; MSIE 4.01; MSN 2.5; Windows 95)	54
Netscape/3.0Gold (X11; I; SunOS 5.5.1 sun4u)	363
Netscape/2.0 (compatible; MSIE 3.01; Windows 95)	227
Netscape/4.04 [en] (Win95; I)	6
Netscape/4.03 [en] (WinNT; I)	897
Netscape/3.01Gold (X11; I; SunOS 4.1.3_U1 sun4m)	225
Netscape/4.03 [en] (Win95; I ;Nav)	37
Netscape/4.01 [en] (Win95; I)	61
Netscape/3.0 (Win16; I)	5
Netscape/4.03 [en] (Win95; I)	4

Table 6.5: Browser and platform types used to access ANNet course

While the majority of the students with access to the course were based at the university itself, data from the remote student and those outside the university, who were given access for evaluation purposes, show successful access from both the United States [81], England [82] and locations within Ireland itself [83][84][85].

6.3.4 Course Usage Versus Exam Performance

As part of the appraisal of the ANNet course an examination into the relationship between student usage of the online ANNet course and their exam grade was performed. Figure 6.1 shows a graph of the number of file accesses per student during the duration of the course against their final exam grades. The number of file accesses gives an overall indication of a students usage of the online course.

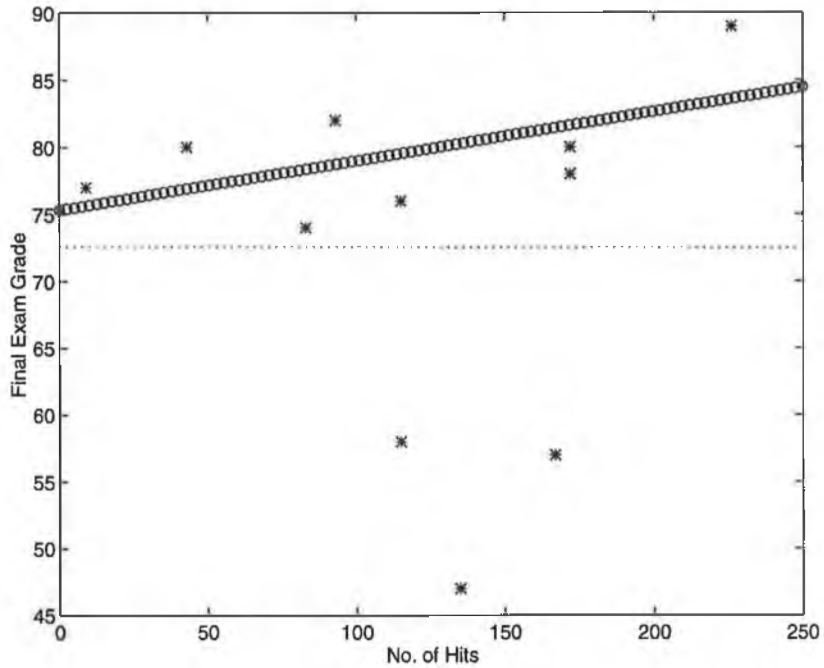


Figure 6.1: Number of file accesses versus exam grade

A best-fit line was fitted to the data (horizontal, dotted line) as seen and was found to yield a correlation coefficient of 0.0025. This shows that for the data set as a whole, there is little or no correlation between the usage rate and the final exam grade. However, if the three outliers at the bottom of Figure 6.1 are removed and the process is repeated then a best fit line, given by the thick line, is obtained.

This set yielded a correlation coefficient of 0.5779. This shows that a trend describing almost 60% of the results was found to be present in the data. Although the statistical set is very small, this trend can be clearly seen, thus showing that usage of the course does contribute to the students final exam grade.

In terms of the remote student, who had the highest usage, his grade *is* almost entirely influenced by the ANNet course as it was his only access to the material, except for recommended texts which were available to students of both formats. This shows that the course provided sufficient material to allow the remote student to grasp the concepts and ideas without the benefit of weekly interaction with the lecturer through the traditional classroom environment, thus meeting the goals of the online course.

6.4 Online Multiple Choice Quizzes

Multiple choice quizzes where provided at the end of each chapter within the ANNet course as outlined in Sections 2.2.2.1 and 5.4.5. Although made available to the students throughout the course, it was not until close to the time of the final exam that students made use of this facility.

There were two ways in which the student could operate the quizzes. Each time the quiz was accessed, a new randomly generated set of questions would appear. However, a second option was available to them. Once the student submitted the quiz for correction, they could use the *BACK* button on the web browser to return to the quiz they had just submitted. This is different to re-accessing the quiz because the same set of questions are present, thus allowing student to re-attempt the questions they answered incorrectly. This process of submitting and re-attempting incorrectly answered questions could be repeated until the students has answered all question correctly. The benefit of this is that students can revise the particular area of a question they answered incorrectly before attempting to answer it again.

Patterns within the answer files showed that students made use of this second approach. This was seen by increasing grades on each submission with only the incorrectly answered question responses being altered. Again an analysis was carried out between the students usage of the quiz facility and their final exam grade.

The correlation coefficient for this data set was found to be 0.2792. Again the statistical set is very small, and made worse by the fact that three of the students did not use the quizzes at all. Thus no hard conclusions can be drawn from these data sets.

Chapter 7

Cost Benefit Analysis

7.1 Introduction

This chapter examines the trade-off between the cost of developing particular features and components of the online ANNet course against the benefit students obtain from them. This *cost* refers to the amount of time, effort and technical expertise required to develop the said features and components. The benefit students gained from them is gauged through their responses to the course review form discussed in Section 6.2. The following section aims to determine the validity of each components inclusion in the ANNet course based on this cost/benefit trade-off. To quantify this value a benefit-to-cost index was used, as shown in Equation (7.1).

$$\text{BTC index} = \frac{\text{Score Achieved in Course Review Form}}{\text{No. of Hours to Complete}} \tag{7.1}$$

where BTC is Benefit to Cost index.

The cost index is the ratio of the students quantitative score for the component against the number of hours work required to develop it. Very low values indicate a high cost in comparison to the benefit student felt they gained from it, while higher values indicate low development costs with high benefits for the students.

7.2 Course Component Development

The components within the ANNet course to be evaluated are the course material itself, the archived mailing list, the Java and MATLAB™ interactive examples and finally the online multiple-choice quizzes. For each component a brief outline of its development, including its estimated time scale, is given and is then followed by a cost benefit analysis as described.

7.2.1 Material Development

Before being developed as an online course, the artificial neural networks module ran for a year as a traditional, classroom based course. As a result of this there was a course structure in place, in terms of the topics being taught and the lecture notes to support these. To transfer the course to an Internet based environment, structural changes also had to be considered in terms of how to layout the course so as to extract maximum benefit from the Internet platform itself so as to provide students with a comprehensive learning environment. Additional material also had to be developed to cover areas and ideas, above and beyond the notes, that would normally be dealt with during the course of a traditional lecturer. These issues are discussed in the following sections.

7.2.1.1 Hypertext and Additional Material

The hypertext structure of the WWW was exploited by the author to layout course material in a layered, non-linear format as described in Section 2.2.2. The use of HTML editors such as that supplied by Netscape [86], makes linking documents via hyperlinks relatively straightforward. The challenge, however, was to place and structure links so as to provide students with the relevant information when and where they need it.

The main advantage of hypertext is that it allows for cross-referencing of topics and material. Foundation material that may be applicable to several different sections can be referenced where necessary. Links can also be provided to material or concepts that are not directly under discussion but are still necessary to the overall understanding. This allows the material to follow a natural path, while still catering for those who do not understand all the ideas presented.

The effort required to explain and add these links is commensurate to that of developing the standard material and can be added at any time throughout the course at the request of students. The ease of adding links to existing documents also minimises the effort involved. It is estimated, by the author, that approximately 450 hours were spent on generating both the standard and additional material for the course. Of this approximately 100 hours were spent on the additional material.

The students on their course review forms rated these hyperlinked, additional notes with a quantitative score of 3.82 on a 1 to 5 scale. This yielded a BTC index of

$$\text{BTC index} = \frac{3.82}{100} = 0.0382 \tag{7.2}$$

This doesn't appear to be very high. However, once completed for the first year of the course the effort in subsequent years is considerably less. It is also worth noting that course material is vital to any course and so is a necessary inclusion regardless of cost.

7.2.1.2 Content Structure and Navigational Aids

Considerable thought and time was employed by the author to devise a course layout that was easy to navigate and was consistent throughout all areas of the course. Several approaches were explored before finally settling on a permanent chapter/section index frame throughout the course. Within the index, topics were arranged in a tree structure thus presenting students with a visual overview of the order and relationship between the topics.

Once decided upon, this layout was easily expandable to all chapters within the course. Students know their location within the course at all times and were always no more than one mouse click away from the main chapter index.

Pages within a chapter were ordered sequentially with in-text links allowing students to jump to additional information where appropriate. Between the index and the sequential forward/backward buttons on each page there is ample provision for students to move about in the course and search for topics. This ease of navigation is essential in any online course, but can be made easier through carefully laying out the course and employing a consistent structure.

It is estimated that the design and creation of the course structure and its navigational system took approximately 90 hours. On the course review forms, navigation of the ANNet course gaine a quantitative score of 4.36. This yields a BTC index of:

$$\text{BTC index} = \frac{4.36}{90} \simeq 0.048 \tag{7.3}$$

Again, this is a one-off cost for the first year. Such a navigational system is essential to an online course and so is deemed necessary regardless of cost. Without it a usable and beneficial course would cease to exist.

7.2.1.3 Weekly Summaries

On the course review form, the weekly summaries received a quantitative score of 4.00 on the 1 to 5 scale. It proved beneficial to both remote and on-campus students. For the on-campus students, they served as a useful reminder of the salient points within each lecture. From the weekly summaries, the remote students were able to determine a suitable pace of study as well as benefiting from the discursive nature of the summaries to provide them with overviews of each section.

In terms of time and effort, the weekly summaries were very cost effective. After each lecture only a small amount time, approximately 45 minutes, was required to recap on the major topics and talking points within that lecture. The summary was then easily distributed to the students via the archived mailing list. Over the duration of the ANNet course there were 12 lecture sessions amounting to a total of 9 hours thus the BTC index was found to be

$$\text{BTC index} = \frac{4.00}{9} = 0.44 \tag{7.4}$$

indicating a good return on the time invested. Thus it is the opinion of the author that the weekly summaries are a worthwhile and beneficial inclusion to the ANNet course.

7.2.2 Archived Mailing List

The communication facility of the ANNet course, in the form of an archived mailing list, required very little effort to set up because a system was already in place within the DCU web server to cater for this (as described in Section 5.6). It proved beneficial both in terms of allowing students to communicate amongst themselves and with the lecturer but also as a means of integrating the on-campus and remote students. The ability to distribute weekly summaries, through the list, was of benefit to both sets. Problems that many students were having could be addressed with a single posting and as could class announcements and details of assignments.

In terms of cost, the archived mailing list is self maintained, except for the approving of subscribers to it, which in itself is just a single email. It is estimated that the approximately 2 hours work was involved in setting up the mailing list itself. On the course review form, the mailing list received a quantitative score of 3.60. Thus the BTC index for the mailing list is;

$$\text{BTC Index} = \frac{3.60}{2} = 1.8 \tag{7.5}$$

Its overall ease of use makes it a useful and necessary component of the ANNet course.

7.2.3 Java Demonstrations

7.2.3.1 Development

The Java Demonstrations included in this course were developed in [1]. Considerable time was invested by their author Mr Paul Keeling both in understanding the Java language and applying it to the development of ANN examples. Every aspect of the example and its environment had to be coded. Also, Java does not have its own built-in graphing capabilities. Thus, these too had to be coded in order to display data in graphical format.

Although, some aspects of the code could be reused, such as the graphing capabilities, each new example had its own characteristics which had to be developed and coded. In terms of development time it is estimated that the time scale per example was 4-6 weeks. Thus for the 6 demonstrations developed the overall time scale is approximately 30 weeks (1125 hours).

7.2.3.2 Benefit Gained

While the examples developed in [1], conveyed clearly the concepts and ideas they were supposed to, the response of students to them was not very favourable. Only 4 out of the 11 who responded to the course review form said that they actually used the Java demonstrations. For the effort and expertise required to develop these demonstrations, the return on investment, in terms of student benefit, was found to be quite poor. Possibly better marketing might encourage more students to try them out, but in terms of time that individual lecturers have to devote to a single course, the development of Java examples proved to be considerable burden. In the normal time frame for course material development for a course, the Java examples require an unreasonable proportion. The BTC index for them was given by;

$$\text{BTC index} = \frac{1.82}{1125} \simeq 0.0016 \qquad (7.6)$$

This is a very poor return on the time and technical investment made for their development. Thus it is the opinion of the author that, in terms of practical development costs, the Java demonstrations impose a unacceptable draw on lecturer resources and so should not be used for the development of the interactive examples.

7.2.4 MATLAB™ Examples

7.2.4.1 Development

MATLAB™ as a computational engine allows users to write their own M-files in either function or script format. Both of these were exploited for the generation of the MATLAB™ interactive examples. Unlike Java MATLAB™ has its own, well documented built-in graphing capabilities, which can generate 2D and 3D plots with relative ease. MATLAB™ M-files can be setup to allow users to enter particular parameters, similar to the Java demonstrations described above. Unlike the Java demos, the interface on the MATLAB™ examples is not as elaborate or colourful, although this can be achieved with additional effort through MATLABs™ demonstration capabilities.

MATLAB™, in conjunction with its many toolboxes, has many engineering functions and operations as executable functions that are ready for use. This considerably reduces the amount of programming required to develop examples.

7.2.4.2 Benefit Gained

An important point of note regarding the MATLAB™ examples in the ANNet course is that they were not originally developed for this purpose. Their original purpose was to generate the plots and graphs for the notes within the classroom ANN course. However, the ease of programming that MATLAB™ offers allowed them to be altered so as to make them interactive and more user friendly. The front end is quite primitive compared with that of the Java demonstrations however it is much simpler and quicker to programme. For lecturers with experience in using

MATLAB™, there is less of a learning curve with regard to the creation of examples. For the examples developed for the ANNet course, each took approximately one day to code, less for some of the smaller ones. Making them more user friendly and interactive took approximately 1 week. Overall the time investment amounted to approximately 110 hours.

MATLAB™ is already a feature within the traditional ANN course as students have to submit assignments that are MATLAB™ based. In this sense students of this particular ANNet course are somewhat biased towards MATLAB™ because of this requirement. This however should not take away from the benefit students gained from them. They rated the MATLAB™ examples with a quantitative score of 4.09 on the 1 to 5 scale. In terms of the BTC index this was found to be:

$$\text{BTC index} = \frac{4.09}{110} \simeq 0.037 \qquad (7.7)$$

This is a vast improvement in terms of return on investment over the Java demonstrations. The files are in script form so students can directly change them, thus allowing them to further explore the mechanics of ANN models. Overall it is the opinion of the author that the MATLAB™ examples are cost effective and beneficial to students and are a valuable addition to the online ANNet course.

7.2.5 Online Quizzes

7.2.5.1 Development

There were two stages of development for the online multiple-choice quizzes. the first was the design and coding of a system for creating and correcting them and the second involved the generation of the questions themselves. As described in Sections 5.4.5 and 5.4.6, the coding of the quiz system was done using Perl. A generic correction system was designed to handle all of the quizzes and a standard format was setup for the generation of the quizzes. This section took approximately 5 weeks to develop and debug. Once this system was in place, focus could then be aimed at generating questions themselves. It is estimated that on average it took 4 hours to generate 30 questions. A further 4 hours was then necessary to develop answers and to place them in the quiz generation script. For the entire course there was a total of 184 questions over 8 quizzes. This is equivalent to approximately 49 hours work. The question were entered in the quiz script as shown in Figure 5.4 (see page 73).

7.2.5.2 Benefit Gained

As mentioned above, once the quiz system was in place the main focus of effort was on the generation of the questions themselves. The technical requirements are all but removed from the process and the lecturer can concentrate of the educational side of the task. When entering the questions into the quiz script, however, some effort is required to convert the questions into HTML and to make slight changes to accommodate the Perl syntax. However a *cut and paste* approach can be adopted after the first question has been entered (see Section 9.4 for future proposals to overcome this problem). In terms of the students perception of the quizzes, they responded positively on the course review form with a quantitative score of 3.45 on the 1 to 5 scale. In terms of the BTC index the online quizzes rated as:

$$\text{BTC index} = \frac{3.45}{49} \simeq 0.07 \tag{7.8}$$

Initially this is not a very high return on the time and effort invested, especially as usage figures were not as high as was hoped, perhaps due to inadequate marketing of the resource. However, the questions are transferable from year to year, so although the initial return on investment in the first year is not high, over the life of the course this will improve greatly. Much of the initial effort was focused on the development of the system itself. After the first year this is no longer an issue and so the cost index for future years will be significantly better.

Overall the effort involved in creating this resource, while requiring considerable thought on the part of the lecturer (in terms of coming up with questions), is deemed to be worthwhile. Student reaction was positive and while no significant relationship between completing the quizzes and exam success was found (Section 6.4, they are still deemed by the author to be a worthwhile component within the online ANNet course.

7.3 Conclusions

Having completed the cost benefit analysis based on the Benefit to Cost (BTC) index the following table was drawn up showing each components results.

File Access Breakdown	
Course Component	BTC Index
Hypertext and Additional Material	0.0382
Content Structure and Navigation	0.048
Weekly Summaries	0.44
Archived Mailing List	1.8
Java Demonstrations	0.0016
MATLAB™ Examples	0.037
Online Quizzes	0.07

Table 7.1: Course components and their corresponding BTC indices

From these figures, it can be seen that the worst scoring components were the Java demonstrations. This is due to, both poor usage and thus rating by the students, and also the large amount of time required to develop them. The hypertext and additional material cost is slightly low. However as described, this can be viewed as a start-up cost that will significantly reduce after the first year. The other components scored comparatively well. Thus, based on these scores and the students responses to them, it is the opinion of the author that they should be included in an online course and, in particular, the ANNet course. The Java demonstrations, while recognising their benefit in terms of conveying conceptual ideas are too time-consuming in their development to be considered for future interactive examples. The MATLAB™ ones are a far more cost effective alternative.

Chapter 8

Maintenance and Reusability

8.1 Introduction

As the ANNet course is a working, commercial product it is vital to its continued success that it can be maintained and upgraded with minimum effort and cost to the lecturer under whose supervision the course lies. This chapter examines the ANNet course with regard to this, and also examines the possibilities of using the course structure and layout as a template for courses of a similar engineering and mathematical nature. The ease with which certain aspects can be improved or modified is considered and where possible guidelines are provided.

8.2 Maintaining the Online ANNet Course

The year-to-year maintenance of the ANNet course can be broken into several sections. These are:

- Year-to-year maintenance (no changes to the course material).
- Adding or modifying course material.
- Adding or modifying the online quizzes.

These are discussed in the following sections.

8.2.1 Year-to-Year Maintenance

In order to reset the ANNet course on a yearly basis the following four tasks must be carried out:

- Old user accounts must be removed and new ones set-up.
- Old statistics files must be archived and new ones created.
- Old quiz results files must be archived and new ones set-up.
- The old archive mailing list must be archived and a new one created.

Once these have been completed the course is ready to be presented for another year. A more detailed description of how to carry out these tasks is provided below.

8.2.1.1 User Accounts

At the beginning of the course, user account have to be set-up to gives students access to the online ANNet course. Essentially this involves deleting the previous year's accounts and replacing them with the current year's. As described in Section 5.5, the usernames are stored in a file called .htaccess, which is located in the *course* directory, and the corresponding passwords are stored in the file .htpasswd which is located in the *conf* directory.

To erase the old usernames, all entries, after the *require user* command, in the .htaccess file are deleted. Similarly the contents of the .htpasswd file are also deleted.

To enter the new accounts the *htpasswd* command, as described in Section 5.5 is used. This is invoked from the command line of the web server and must be repeated for each individual user account. The web server is accessed via a *telnet* session. The telnet program within Windows 95 can be used for this purpose. The web server name is *khumbu.eeng.dcu.ie.* To gain access the username and password of the server account has to used (this is obtained from the web administrator).

8.2.1.2 Statistics Files

As described in Section 5.4.7, all file accesses within the ANNet course are logged. Each chapter has its own file to which its statistics are logged. To reset these, the old files can be renamed (e.g. stats_chpt1.txt could be renamed to stats_chpt1-97.txt). This allows the lecturer to maintain record about the course over several years. Once this is done then new files are created to replace them. The new files have to be saved with exactly the same names as the previous ones and in the same place. This can be done using the FTP package, WS_FTP [87]. This package provides an interface to the web server and allows for the uploading/downloading of files. renaming of files, etc. The new files can be created in any text editor, by saving a blank text file with the appropriate name and then uploading it to the web server.

After this is done the statistics program will operate as was, collecting all the usage details for the current year.

8.2.1.3 Quiz Results files

To rest the online quiz files the same process as for the statistics files is employed. The results files are found in the main *annet* directory, and take the form of chpt1_quiz_results.txt. These old results files can be renamed and the new ones have to be saved in the same place and with the same names as their predecessors. The quiz system will then log the results as normal.

8.2.1.4 Archived Mailing List

To reset the archived mailing list, the assistance of the web administrator is required. He/she has the authority to create new lists and archive old ones. Once this has been done, the list will operate as previously with all new subscribers having to be approved by the list owner, i.e. the course lecturer. This is carried out via email correspondence with the mailing list package. When a user tries to subscribe to the list an email is sent to the list owner giving details of the potential subscriber and of what to do to subscribe them. The owner then uses their password and follows these instructions. It is a relatively straightforward process with instructions provided for each step.

8.2.2 Adding or Modifying Course Material

The most important element of any course is its notes and course material. Therefore it is vitally important to the continued success of the ANNet course, that material can be added, removed or updated by the lecturer with relative ease.

8.2.2.1 Notes and Mathematical Equations

As mentioned in Section 5.2.4, there are many freely and commercially available editors and converters that allow instructors to generate HTML material or to convert, from existing formats, to HTML. The problem however, as underlined in Section 5.3, arises when dealing with mathematical equations and symbols. The approach adopted and suggested by the author, until MathML comes online, is to use image files.

Course material developed in LaTeX, can be converted to HTML using the *latex2html* conversion package. This converts the LaTeXformatting within the document to HTML. Any equations or symbols within the LaTeX document are converted to image files. This approach is suited to documents that are already in LaTeXformat. Microsoft WORD also provides a facility to convert WORD documents to HTML. The equations and symbols are also converted to images files.

For the generation of images Paint Shop Pro [88] was used by the author. This package can handle many image types and has many useful facilities (e.g. cropping, editing and resizing of images). These images are then added to the HTML document as described in Section 5.3.1.

8.2.2.2 Directory Structure

To ensure that the course material stored on the web server is structured and easily maintained, the author has set-up a directory tree as shown in Figure 8.1.

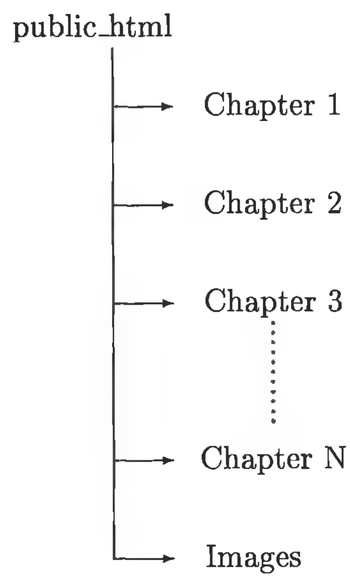


Figure 8.1: ANNet directory structure

Each chapter within the course has its own directory in which all the HTML documents relating to it are placed. Any image files, relating to all chapters, are placed in a single directory, aptly named *Images*. The reasoning for this is that some image files may be applicable to several or all of the chapters (i.e. the navigational buttons) and so can all the called from the one place. The alternative of placing images in separate chapters could result in multiple copies of files, thus taking up unnecessary space on the web server.

When a new chapter is to be created it is simply a matter of creating a directory for it and placing its files within. Once this is done the chapter/section index has to be altered to cater for the newly added sections or chapters. To do this the file *begin.shtml* in the *intro* directory must be altered. For each chapter entery the following lines are added.

```
<P><IMG SRC="red_bl.gif">
<A HREF="Chapter2.html" TARGET="begin">
Chapter name
</A>
</P>
```

To add new sections to the section index go to the directory of the chapter you wish to modify and open the *sections.shtml* file. Add the lines

```
<P><IMG SRC="../images/green_bl.gif">
<A HREF="filename.shtml" TARGET="main1">
Section name
</A>
</P>
```

for each new section. To add sections with subsections below them use the lines

```
<P><IMG SRC="../images/green_bl.gif">
<A HREF="filename.shtml" TARGET="main1">Section name</A>
<UL>
<P>
<IMG SRC="../images/red_bl.gif" HSPACE=4 BORDER=0 HEIGHT=14 WIDTH=14>
<A HREF="filename1.shtml" TARGET="main1">Subsection name</A>
<P>
<IMG SRC="../images/red_bl.gif" HSPACE=4 BORDER=0 HEIGHT=14 WIDTH=14>
<A HREF="filename2.shtml" TARGET="main1">Subsection name</A>
</UL>
</P>
```

8.2.3 Online Quizzes

The online quizzes within the ANNet course, like the course material itself, have to be flexible to changes and additional content. Thus the software had to be designed with this in mind. As outlined in Section 5.4.6, the correction facility is common to all quizzes once the answer file has been created, thus requiring no changes for additional quizzes that may be developed. A similar process is in place for actually generating the quizzes. However, it does require changes from quiz to quiz and so is not as generic as the correction process.

To add new question or to create a new quiz the following two tasks must be completed.

- Modify the test script to include new questions or to completely generate a new one.
- Modify or create an answer file

8.2.3.1 Adding Questions To An Existing Quiz

To add new questions to an existing quiz, there are two changes that have to be made to the quiz script

1. Add the question itself.
2. Change the number of questions within the script.

Adding the Question

As described in Section 5.4, each question is entered as a subroutine within the script. Consider the question in Figure 8.2 (see page 112). It is similar to that in Figure 5.4 (see page 73), except it is in a generic format.

```
sub question_9
{
print "<FONT SIZE=+1><P>
        The question goes here .... :
        </P><OL>";
print "<LI><INPUT TYPE=\"radio\" NAME=\"q9\" VALUE=\"a\">
        Option A</LI>
        <LI><INPUT TYPE=\"radio\" NAME=\"q9\" VALUE=\"b\">
        Option B</LI>
        <LI><INPUT TYPE=\"radio\" NAME=\"q9\" VALUE=\"c\">
        Option C</LI></OL></FONT>"; }
```

Figure 8.2: Generic form of multiple-choice question in Perl script

This is essentially HTML with some Perl syntax around it. The *NAME* tag within each of the optional answer lines refers to the question number and should be changed accordingly. The *VALUE* tags refer to the individual optional answers. For example if there were 4 alternative answers then a fourth line similar to the existing ones would be entered with the VALUE tag set equal to *d*. Both of these tags are important so that the correction process will know which question it is dealing with and which answer within that question has been chosen.

Informing the Quiz Script of the Additional Question

Once the question itself is added, some small changes have to be made to the script. As the questions are randomly selected each time the quiz is invoked, the quiz has to be informed of the additional question, so as to include it in the selection process. Essentially this involves adding an extra line to an if statement within the script and increasing the *number of questions* variable within the script. This is done as follows:

- Increase the range of the random number generator by the number of questions added. i.e. in the section labelled "Generate random numbers to choose

```
srand ( time() ^ ($$ + ($$ << 19)) );
$j = int(rand 23) +1; # generate integer btw 1 and 19

@rand_nums[0] = $j;
for($i = 0; $i < 10;  $i++)
{
       $j =  int(rand 23) +1;
       for ($a = 0; $a < $i; $a++)
       {
            if ( @rand_nums[$a-1] == $j )
            {
                   $j =  int(rand 23) +1;
                   $a = 1;
            }
       }
       @rand_nums[$i] = $j;
}
```

Figure 8.3: Section of tutorial script to generate random question numbers

random questions" of the tutorial script (as shown in Figure 8.3), change the value within the term $int(rand23)$ to the correct number of questions, i.e. the original value plus the number of new questions added. This term appears 3 times within the section and the value must be changed each time.

- Next a slight modification to the *if* statement within the section labelled "Generate questions corresponding to random numbers" of the tutorial script is required. The purpose of this is to include the new questions in the scripts selection process. For example, the end of th if loop reads:

```
elsif($question_number == 19) {@question[$i] = "&question_19;";}
elsif($question_number == 20) {@question[$i] = "&question_20;";}
elsif($question_number == 21) {@question[$i] = "&question_21;";}
elsif($question_number == 22) {@question[$i] = "&question_22;";}
```

```
else{@question[$i] = "&question_23;";}
}
```

If a 24th question were to be added the loop would be changed to;

```
elsif($question_number == 19) {@question[$i] = "&question_19;";}
elsif($question_number == 20) {@question[$i] = "&question_20;";}
elsif($question_number == 21) {@question[$i] = "&question_21;";}
elsif($question_number == 22) {@question[$i] = "&question_22;";}
elsif($question_number == 23) {@question[$i] = "&question_23;";}
else{@question[$i] = "&question_24;";}
}
```

Now the script is ready to run with the new questions added.

8.2.3.2 The Answer File

The answer file for a quiz is only accessed or edited when

- new questions are added to a quiz, or
- when a new quiz is generated

Adding Answers to Newly Added Questions

When a new question is added, as described in Section 8.2.3.1, the corresponding correct answer has to be added to the answer file. The answer files for all the quizzes are located in the main *annet* directory. The format in which questions are added to the answer file is outlined in Section 5.4.6. New question answers are added in exactly the same format.

Creating a New Answer File

The process of creating a new answer file is very straightforward. The question answers are entered in the text file as outlined in Section 5.4.6 and saved with an appropriate name in the same location as the other answer files.

The only difference is that the name and location of the text file have to be entered in the tutorial script. This appears at the top of the quiz script in the following line

```
print "<INPUT type=\"hidden\" NAME=\"answer_file\"
        VALUE=\"/user/mailing/annet/answers_chpt1.txt\">";
```

The file name shown is replaced by the new file name. Once completed the new quiz is ready for use.

8.2.3.3 To Create a New Quiz

The simplest way to create a new is to save an existing one under a new name and modify it for the new quiz. This would essentially involve:

- Changing the quiz name and details.
- Adding the new questions as described in Section 8.2.3.1.
- Informing the quiz script of the correct number of question, as described above.
- Create an answer file

Once this is carried out the file is saved to the *cgi-bin* directory. Before it can be accessed, the permissions of the file have to be set. This is done at the command line of the web server. Login to the course directory as described in Section 8.2.1.1 and type the following two commands (pressing enter after each one).

```
chmod a+x
chmod go-w
```

8.3 Template for New Courses

While the ANNet course was developed with a view to teaching ANN theory, much of the structure and its components can be used as a foundation upon which to develop other courses. A solid and easily expandable directory structure was developed to facilitate storing the material (HTML documents). It optimises space and allows for incremental development of sections and material.

Many of the components such as the online quizzes and communication features, i.e. the mailing list can be incorporated to any online course. The use of MATLAB™ examples to convey conceptual ideas can be adopted to many engineering and science based subjects. The monitoring facilities can also be adopted by any online course.

It is the opinion of the author that the ANNet course while specific in its subject area is also flexible enough to be used as a template for other courses, particularly in the area of engineering and science.

Chapter 9

Conclusions and Recommendations for Future Work

9.1 Introduction

This thesis has dealt with the growing area of computer aided learning and, in particular, the field of Internet-based learning. The subject of artificial neural networks was taken as a practical example and a functional online course to teach this topic was developed. This chapter presents the authors findings with regard to CAL and in particular Internet CAL and also the suitability of ANNs to this medium. Future recommendations are set-out with regard to the course itself, detailing areas and features within the course that can be built upon.

9.2 CAL: The WWW Based Approach

As the results of this thesis show, CAL has many benefits to offer education, in particular the fields of engineering and sciences who already rely heavily on computers for many of their applications. The flexibility and accessibility of the Internet further increase the potential role CAL has to play within the education system, in particular with regard to distance education and continued professional development courses as described in Section 2.3.2. Through this Internet medium, course material can be presented to students on a platform that incorporates many other media, such as graphics, sound, audio and video. Students can draw on the Internet's vast resources to find additional material and real world applications of these subjects all in one location (i.e. their PC). Thus the author feels that education has a lot to benefit from the WWW as a platform for both supplemental material and also fully operational courses.

At present, however, the author does acknowledge that there are restrictions with regard to speed of access and material development etc. that currently limit some of these qualities. In the case of the ANN course, and indeed engineering and science subjects in general, the lack of true mathematical capabilities is a distinct disadvantage. While lecturers may be keen to convert material to the WWW delivery mode, their limited experience in material development within this environment may discourage them. Advances in these areas, while currently in progress, are necessary before the full power the Internet has to offer to engineering education can be realised.

Separate to the development issues associated with online courses are the copyright and material ownership rights of these courses. This is still a very grey area for most colleges due to the fact that the whole concept is relatively new. Fears that lecturers may lose the rights to such courses and their material, must be addressed.

Despite the developmental and legal concerns associated with the WWW mode of educational delivery, it is the opinion of the author that the Internet offers many advantages to the educator and will be a force within education, in particular engineering education in the near future.

9.3 CAL and Artificial Neural Networks

The field of artificial neural networks was found to be a very suitable subject for an Internet based mode of delivery. Core ideas and concepts traversed many sections within the course, proving ideal for the hypertext structure of the WWW. The modular layout of the ANN course also lends itself to this platform. Topics within the course can be laid out as individual blocks of ideas and hence can be referenced from anywhere within the course.

The iterative algorithms that characterise much of ANN theory are ideally suited to computer simulations, while graphical representations of these ideas, through the use of plots and interactive examples, proved very beneficial to their overall understanding.

Overall, it is the opinion of the author that ANNs and subjects of a similar computational and structural background, are well suited to the mode of delivery offered by the WWW and as such this medium should be considered by those developing new courses, or upgrading existing ones, as a viable supplement and/or alternative to traditional classroom methods.

9.3.1 The ANNet Course

As a practical example of an Internet based course in ANNs, the ANNet course was deemed was to be a success. A fully functional course environment was developed, complete with interactive examples, self-test facilities, student monitoring facilities, communication facilities and a comprehensive set of notes. Although there was only a single remote student taking the course, his success with the course, in terms of his exams results, is encouraging for the future. The use of the course as supplemental material to the on-campus students also proved successful based on students' responses (see Chapter 6).

Overall the application of ANNs to the Internet CAL environment proved successful, receiving positive feedback from the students who undertook the ANNet course (see

Chapter 6). The ANNet course developed, while successful in its own right, can also be used as a template upon which to base other courses of a similar nature.

9.4 Future recommendations

With regard to the ANNet project developed within this thesis, some recommendations are laid down by the author. These refer to several aspects of the ANNet course and are intended to further improve the learning experience of the student undertaking it. They are laid down in the following sections with a brief description of the motivation for them.

9.4.1 Course Material

The findings of Chapter 6 show that students found the addition material within the ANNet course very beneficial. This additional material was primarily focused on the first four chapters of the course. It is the recommendation of the author that this material be extended to cover all aspects of the course.

A cheaper alternative to publishing course material entirely in WWW format might be to provide downloadable versions of the core notes and to only publish additional material and examples. This approach, while removing the hypertext, nonlinear element from the course still allows for all the remote access and interactive qualities to be included. Development costs would be reduced and the requirements for students to spend long period of time logged on to the course to access the material would also be minimised.

9.4.2 Online Quizzes

While the author has developed a multi-choice quiz facility for the course, some recommendations are presented with regard to improving both student benefit and ease of development on the part of the lecturer.

1. With the quiz facility itself, the author recommends the development of a front end that will allow lecturers to add and develop quizzes without needing to access the underlying software, as is currently the case. This would involve a facility whereby lecturers could enter questions in text format, which would then be dealt with by the quiz software and added automatically to the relevant areas.

2. In terms of feedback to students, the author recommends that as well as informing students of the questions they answered incorrectly, they should also receive information regarding where in the course material the material or concepts involved can be found. This would reduce the temptation for students to *guess* answers and encourage them to further examine the areas in question.

References

[1] Keeling, P., *"Internet-based Computer-Aided Learning for Artificial Neural Networks"*, M.Eng Thesis, School of Electronic Engineering, DCU, September, 1997

[2] Galvin, G., *"Computer Aided Learning Tool"*, B.Eng Report, School of Electronic Engineering, DCU, April, 1997

[3] Ruhr University Bochum. Virtual Control Lab 2 [online]. Available from: http://www.esr.ruhr-uni-bochum.de/VCLab [Accessed 10 August 1997]

[4] Starr, R.M. *"Delivering Instruction on the World Wide Web: Overview and Basic Design Principles"*, Educational Technology, May-June 1997

[5] Wilson, T. and Utecht, G. *"The Internet at Eagan High School"*. T.H.E. Journal, April 1995, 22(9), 75-79.

[6] Zurada, J.M., *"Introduction to Artificial Neural Systems"*, West Publishing Company, p xiv, p54

[7] Haykin, S. *"Neural Networks: A Comprehensive Foundation"*, Prentice-Hall Inc, 1994, p1, p 36, p45, p142, p236

[8] Fausett, L., *"Fundamentals of Neural Networks: Architectures, Algorithms and Applications"*, Prentice-Hall Inc, 1992, p 3, p26

[9] Penfield, P. and Larson, R.C. *"Education via Advanced Technologies"*, IEEE Transactions on Education, Vol 39, no 3, August 1996

[10] Lee, P.M. & Sullivan, W.G. *"Developing and Implementing Interactive Multimedia in Education"*, IEEE Transactions on Education, Vol 39, no 3, August 1996

[11] Hansen, E. *"The role of interactive video technology in higher education: Case study and proposed framework"*, Educational Technology, Sept 1990, pp 13-21,

[12] Lai, L. L. *"Computer Assisted Learning in Power System Relaying"*, IEEE Transactions on Education, Vol 38, No. 3, August 1995

[13] Myers, D.K. *"Interactive Video: A chance to plug the literacy leak"*, Industry Week, no 239, pp 15-18, April 1990

[14] Cockerton, T. and Shimell, R. *"Evaluation of a hypermedia document as a learning tool"*, Journal of Computer Assisted Learning(1997) 13, 133-144

[15] Tolhurst, D. *"Hypertext, Hypermedia, Multimedia Defined"*, Educational Technology, March-April 1995, p21

[16] Kommers, P.A.(1996) *"Hypermedia and multimedia concepts: Definitions."* In P.A Kommers, S. Grabinger and J.C. Dunlap. *"Hypermedia Learning Environments: Instructional Design and Integration."* Lawerance Erlbaum Associates, Publishers. Mahwah, New Jersey.

[17] Mengel, S.A. and Adams, J.A. *"The need for Hypertext Instructional Design Methodology"*, IEEE Transactions on Education, Vol 39, No. 3, August 1996

[18] Gagne, R. and Wager, W. and Rojas, A. *"Planning and authoring computer assisted instruction lessons"*, Educational Technology, 1981, 21, pp 17-26

[19] Thomas, P. and Carswell, L. and Price, B. and Petre, M. *"A holistic approach to supporting distance learning using the Internet: transformation, not translation."*, British Journal of Educational Technology, Vol 29, No.2, 1998, p149

[20] Galbreath, J. *"The Internet: Past, Present and Future"*, Educational Technology, November-December 1997, pp 39-45

[21] Harger, H.O. *"Teaching in a Computer Classroom with a Hyperlinked, Interactive Book"*, IEEE Transactions on Education, Vol 39, No. 3, August 1996, p327

[22] Kearsley, G. and Lynch, W. and Wizer, D. *"The Effectiveness and Impact of Online Learning in Graduate Education"*, Educational Technology, November-December 1995, pp 35-42

[23] Nua Internet Surveys. ZD Market Intelligence [online]. Available from: http://www.nua.ie/surveys/index.cgi?service=view_survey&survey_number=774&rel=n [Accessed 8 July 1998]

[24] Sloane, A. *"Learning with the Web: Experience of using the World Wide Web in a learning Environment"*, Computers in Education, Vol 28, No. 4, pp207-212, 1997

[25] University of Michigan. Distance Learning: Planning Considerations and Options [online]. Available from: http://www.oit.itd.umich.edu/reports/DistanceLearn/sect5.1-4.html [Accessed 14 July 1998]

[26] Springhall, N.A. and Springhall, C.C. and Oja, S.N. *"Educational Psychology: A Developmental Approach"*, McGraw-Hill Inc., 6ed, 1994.

[27] Aktan, B. and Bohus, C.A. and Crowl, L.A. and Shor, M.H. *"Distance Learning Applied to Control Engineering Laboratories"*, IEEE Transactions on Education, Vol 39, No. 3, August 1996

[28] Hiltz, S.R., *"The Virtual Classroom: Learning without Limits via Computer Networks"*, Ablex Publishing, Norwood, New Jersey, 1994.

[29] The Microsoft Corporation [online] Available from: http://www.microsoft.com [Accessed 12 Nov. 1997]

[30] Goldberg, M.W. and Salari, S. *An Update on WebCT (World-Wide-Web Course Tools) - a Tool for the Creation of Sophisticated Web-Based Learning Environments*, Papers Related to WebCT. [online] Available from: http://homebrew1.cs.ubc.ca/webct/papers/nauweb/full-paper.html [Accessed 26 May, 1998]

[31] WebCT, More Information about WebCT [online] Available from: http://homebrew1.cs.ubc.ca/webct/intro/ [Accessed 28 May, 1998]

[32] WebCT, World Wide Web Course Tools [online] Available from: http://www.webct.com/webct/get/pricing.html [Accessed 15 July, 1998]

[33] WBT Systems, Products White Paper [online] Available from: http://www.wbtsystems.com/products/whitepa1.htm [Accessed 28 May, 1998]

[34] PC Week Lab, Labs on Site. [online] Available from: http://www8.zdnet.com/pcweek/reviews/ibt.html [Accessed 28 July, 1998]

[35] Nakayama, K. and Chigawa, Y. and Hasegawa, O., *"Handwritten Alphabet and Digit Character Recognition Using Feature Extracting Neural Network and Modified Self-Organising Map"*, International Joint Conference on Neural Networks, Baltimore, Maryland, June 7-11, 1992, Vol. 2, pII-419

[36] Young, S.S. and Scott, P.D. and Nasrabadi, N.M., *" Object Recognition Using Multi-Layered Hopfield Neural Networks"*, IEEE Transactions on Image Processing, Vol 6 No. 3, March 1997

[37] Chiou, G.I. and Hwang, J.N., *"A Neural Network-Based Stochastic Active Contour Model (NNs-SNAKE) for Contour Finding of Distinct Features"*, IEEE Transactions of Image Processing, Vol 4, No. 10, October 1995

[38] Ringwood, J.V., *"An Introduction to Artificial Neural Networks"*, M.Eng Course Notes, Dept. of Electronic Engineering, DCU, 1997-1998

[39] McColloch, W.S. and Pitts, W., *"A Logical Calculus of the Ideas Immanent in Nervous Activity"*, Bulletin of Mathematical Biophysics, Vol. 5, pp115-133, 1943

[40] Hebb, D.O., *"The Organisation of Behaviour: A Neuropsychological Theory"*, New York: Wiley, 1949

[41] Rosenblatt, F., *"The Perceptron: A Probabilistic Model for Information Storage and Organisation in the Brain"*, Psychology Review, Vol. 65, 1958

[42] Akay, M. and Akay, Y.M. and Welkawitz, W., *"Neural Networks for the Diagnosis of Coronary Artery Disease"*, International Joint Conference on Neural Networks, Baltimore, Maryland, June 7-11, 1992, Vol. 4, pIV-235

[43] Cloarec, Guy-Michel, *"Multistep Ahead Prediction of the Sunspot Series using Hybrid Neural Network Models"*, Research Report EE/JVR/97/3. School of Electronic Engineering, Control Systems Group, Dublin City University, 1997.

[44] White, J., *"A Genetic Adaptive Neural Network Approach to Pricing Options: A Simulation Analysis"*, Journal of Computational Intelligence in Finance, Vol.6, No.2, March/April 1998

[45] Chen, S. and Billings, A., *"Neural networks for nonlinear dynamic system modeling and identification"*, Int. J.Control 56(1992)pp.319-346

[46] Narendra, K.S. and Parthasarathy, K., *"Identification and Control of Dynamic Systems Using Neural Networks"*, IEEE transactions on Neural Networks, Vol. 1, pp 4-27, 1990

[47] Lisboa, P.G.J, *"Neural Networks - Current Applications"*, Chapman and Hall Publishers, 1992, p2

[48] Lipmann,R.P., *"An introduction to Computing with Neural Nets"*, IEEE ASAP Magazine, April 1987

[49] Werbos, P.J., *"Beyond regression: New tools for prediction and analysis in behaviour science*, Ph.D thesis, Harvard University, Cambridge, 1974

[50] Rumelhart, D.E. and McClelland, J.L. and the PDP Research Group, *"Parallel Distributed Processing (PDP): Exploration in the Microstructure of Cognition*, Vol. 1. The MIT Press, Cambridge, MA, 1986

[51] Powell, M.J.D, *"Radial basis functions for multivariate interpolation: A review"*, In IMA Conference on Algorithms for the Approximation of Functions and Data, pp143-167, RMCS, Shrivenham, UK, 1985

[52] Broomhead, D.S. and Lowe, D., *"Multivariate functional interpolation and adaptive networks"*, Complex Systems 2, pp 321-355, 1988

[53] Hush, D.R. and Horne, B.G., *"Progress in Supervised Neural Networks: Whats new since Lipmann ?"*, IEEE Signal Processing Magazine, January 1993

[54] Cover, T.M., *Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition"*, IEEE transactions on Electronic Computers, Vol 14, p326-334, 1965

[55] Hopfield, J., *"Neural Networks and Physical Systems with Emergent Collective Computational Abilities"*, Proceedings of the National Academy of Science, Vol. 79, pp2554-8, 1982

[56] Kohonen, T., *"The Self-Organizing Map"*, Proceedings of the IEEE, Vol. 78, No. 9, September 1990

[57] Kertesz, A., *"Localisation in Neuropsychology"*, New York Academic Press, 1983

[58] Patterson, D.W., *"Artificial Neural Networks: Theory and Applications"*, Prentice-Hall Inc., 1996

[59] Gray, R.M., *"Vector Quantization"*, IEEE ASAP Magazine, Vol. 1, 4-29, 1984

[60] Box, G.E.P. and jenkins, G.M., *"Time Series Analysis: Forecasting and Control"*, Holden-Day, 1976

[61] Danell, R., *"Use of Neural Networks for Data Analysis in Chemistry"*, Graduate Student, University of North Carolina at Chapel Hill. email: rdanell@email.unc.edu

[62] HyperMail Documentation [online] Available from: http://www.nnic.noaa.gov/hypermail.html [Accessed September 1, 1998]

[63] MATLAB™, Mathworks [online] Available from: http://www.mathworks.com [Accessed August 21, 1998]

[64] Hoffman, P.E., *"The Internet: Instant Reference"*,SYBEX Inc., 1995

[65] Network Wizards, Internet Domain Survey, July 1998 [online] Available from: http://www.nw.com/zone/WWW/report.html [Accessed 17 Aug, 1998]

[66] Hahn, H. and Stout, R., *"The Internet: Complete Reference"*, Osbourne McGraw-Hill, 1994

[67] Counseil Europeen pour la Recherche Nuclaire (CERN) [online] Available from: http://www.cern.ch/ [Accessed 17 August, 1998]

[68] World Wide Web Consortium, [online] Available from: http://www.w3.org/ [Accessed 18 August, 1998]

[69] World Wide Web Consortium, HTML 4.0 Specification, [online] Available from: http://www.w3.org/TR/REC-html40/ [Accessed 19 August, 1998]

[70] Daconta, M.C., *"Java for C/C++ Programmers"*, John Wiley & Sons, Inc., 1996

[71] Kopka, H. and Daly, P.W., *"A Giude to $\LaTeX 2_\varepsilon$: Document Preparation for Beginers and Advanced Users"*, Addison-Wesley, 1996

[72] Miller, P. Electronic Publishing Editor, *"Web Sights"*, IEEE Spectrum, June 1998

[73] World Wide Web Consortium, MathML Draft [online] Available from: http://www.w3.org/TR/PR-math/ [Accessed 20 August, 1998]

[74] World Wide Web Consortium, Amaya, Browser [online] Available from: http://www.w3.org/Amaya/ [Accessed 20 August, 1998]

[75] Wall, L and Christiansen, T and Schwartz, L., *"Programming Perl"*, O'Reilly & Associates Inc., 1996

[76] Husain, K., *"Perl 5 Unleashed"*, Sams Publishing, 1996

[77] Denova, C.C, *"Test Construction for Training Evaluation"*, Van Nostrand Reinhold Company, New York, 1979

[78] *"MATLAB: A Technical Language of Computing"*, The Mathworks Inc., 1997

[79] *"The Student edition of MATLAB: A Technical Language of Computing"*, Prentice-Hall Inc., 1997

[80] Hague, P., *"Questionnaire Design"*, Clays Ltd., St. Ives Plc., 1993

[81] Computational Sciences Division, Boeing, Seattle. Server Identity: www-fw-proxy3.boeing.com

[82] University of Strathclyde, Glasgow. Server Identity: elizabeth.eee.strath.ac.uk

[83] Dept. of Food Engineering, UCC. Server Identity: fe21.ucc.ie

[84] Robert Manley; Institute of Technology, Dundalk via Ireland Online: Public Internet Access Service Provider Server Identity: dialup-007.dundalk.iol.ie

[85] Niall Mollen: ANNet student via Indigo: Public Internet Access Service Provider Server Identity: ts02-104.dublin.indigo.ie

[86] Netscape Netcentre [online] Available from: http://home.netscape.com/browsers/index.html [Accessed September 1, 1998]

[87] Ipswitch WS_FTP FTP, Software[online] Available from: http://www.ipswitch.com/ [Accessed September 8, 1998]

[88] JASC Software, Pain Shop Pro, [online] Available from: http://www.jasc.com/psp.html [Accessed September 15, 1998]

Appendix A

ANNet Course Contents

Chapter 1 :- Introduction to ANNs

- Biological Inspiration
- Brief History of ANN Development
- Types of ANN
- Perspectives on ANNs
- Learning in ANNs
- Some Pros and Cons
- Applications of ANNs
- Introduction to the Matlab ANN Toolbox
- Multiple Choice Quiz

Chapter 2 :- Simple Neural Networks and Pattern Classification

- The McColloch-Pitts Neuron and the Peceptron
- Hebbian Learning

  Threshold Neuron with Hebbian Learning

  Continuous Neuron with Hebbian Learning
- Perceptron Learning Rule

- Perceptron Convergence Theorem
- Ideal Learning Rate
- Classification and Linear Separability
- Application to Common Boolean Functions
- The Adaptive Linear Combiner

  Example: One Step Predictor

  Numerical Solution Using Gradient Search
- The LMS Algorithm

  One Step Predictor: Revisited
- The Adaline

  Classification Example: Using the Adaline and LMS Algorithm

  Classification Example: Using Perceptron Rule

  Conclusions and Comments
- The Delta Learning Rule

  Derivation of Delta Rule

  Evaluation of Non-Linearitiy Derivatives

  Classification Example Using Delta Rule: Unsolvable Problem

  Classification Example Using Delta Rule: Solvable Problem
- Multiple Choice Quiz

Chapter 3 :- Multi-Layer Networks (MLPs)

- Introduction to MLPs

  Function Approximation Using MLPs

  Number of Hidden Layers
- Error Backpropagation

- The Ex-OR Problem
    - Analytical Solution
    - Adaptive Solution
- Improving the Learning Algorithm
    - Momentum Method
    - Adaptive Learning Rate
    - Pattern and Batch Update
    - Higher Order Techniques
    - Scaling of Training Data
- MLP Architecture Optimisation
    - Complexity Regularisation
    - Sensitivity Analysis
- Multiple Choice Quiz

Chapter 4 :- Radial Basis Function Networks (RBF)

- Radial basis Function Networks
- RBFs for Classification Problems
    - The Ex-OR Problem (Again !)
- RBFs for Function Approximation
    - Improvement of Generalisation Property
    - reduction in Number of Neurons
    - Addition of Bias Term
- The Ex-OR Problem
- Learning Strategies for RBF Networks
    - Strategy 1: Fixed Centres

Strategy 1: Self-Organised Selection

Strategy 1: Adaptive (Supervised) Selection

- Java Demonstration

- Multiple Choice Quiz

Chapter 5 :- Dynamic Neural Networks

- Dynamic Neural Networks

  Implementation Methods

  Applications Tapped Delay Line Neural Networks

  IIR Version

- FIR Multi-Layer Perceptron

- Discrete Time Recurrent Neural Networks

  Example of a Recurrent Training Algorithm - Linear Case

  Points to Note

- Multiple Choice Quiz

Chapter 6 :- Self Organising Feature Maps (SOFM)

- Self-Organising Feature Maps

- Architecture

- Topological Map

- Self Organising

- Basic Competitive Learning

  Example: Vector Quantisation

  Proof of Convergence

  Example of Vector Quantisation

- A Self Organising Map
    - Selection of Best-Matching Cell
    - Adaptation of the Weight Vectors
    - SOFM Example
- Comments
- Java Demonstration
- Multiple Choice Quiz

Chapter 7 :- Hopfield Networks

- Hopfield Networks
- Linear Analysis
- Problem Statement
- Dynamics of Hopfield Network
- Pattern Storage
    - Diagonal Weights
    - Storage of Multiple Patterns
    - Example
- Storage Capacity
- Spurious States
- Further Comments
- Java Demonstration
- Multiple Choice Quiz

Chapter 8 :- Data Preprocessing

APPENDICES

- Application of Neural Networks
- Application Procedure
- Time Series Modeling
- Time Series Modeling

Appendix B

Neural Networks Online Course Evaluation Form

	Very bad	Bad	Neutral	Good	Very good
How would you rate the presentation of the online course with regard to ease of learning ?	1	2	3	4	5
How would you rate your ability to navigate your way through the online course ?	1	2	3	4	5
How would you rate the online course in terms of access (i.e. delay in downloading pages for viewing) ?	1	2	3	4	5
Did you find the material in the online course sufficient for understanding the topics presented ?	1	2	3	4	5
Did you use the MATLAB files provided for the various examples (1 = none, 5 = all) ?	1	2	3	4	5
If yes, how would you rate the benefit you gained from them ?	1	2	3	4	5
Did you use the Java demonstrations provided for the various examples (1 = none, 5 = all) ?	1	2	3	4	5
If yes, how would you rate the benefit you gained from them ?	1	2	3	4	5

How would you rate the online quizzes as an indicator of your progress/understanding of the course material ?	1 2 3 4 5
How would you rate the archived mailing list in terms of its benefit to the overall learning experience ?	1 2 3 4 5
How would you rate the online course in terms of access (i.e. delay in downloading pages for viewing) ?	1 2 3 4 5
Did you find the weekly summaries beneficial ?	1 2 3 4 5
Did you find the downloadable postscript note beneficial ?	1 2 3 4 5
Did you find the extra links and explanations within the online course beneficial ?	1 2 3 4 5

What changes would you like to see made to the online course in order to improve the presentation of the material ?

Are there any areas within the online course that you feel could be better explained or covered in more detail ?

What other tools or features would you like to see in the online course in order to improve the overall learning experience ?

Other Comments

Appendix C

Chapter Statistics Files - extracts

Username, Page Accessed, Date and Time, Remote Host, Browser Type

```
plawless, example_2.shtml, Thu Mar  5 21:47:49 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, summary.shtml, Thu Mar  5 22:17:56 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, toolbox.shtml, Thu Mar  5 22:18:05 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, summary.shtml, Fri Mar  6 09:18:49 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, toolbox.shtml, Fri Mar  6 09:18:53 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, example_2.shtml, Fri Mar  6 11:39:40 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, summary.shtml, Sat Mar  7 12:45:28 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, boi.shtml, Sat Mar  7 12:46:06 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, history.shtml, Sat Mar  7 12:46:22 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, types.shtml, Sat Mar  7 12:46:26 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, learning.shtml, Sat Mar  7 12:46:34 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, procon.shtml, Sat Mar  7 13:01:26 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, learning.shtml, Sat Mar  7 13:02:48 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, applic.shtml, Sat Mar  7 13:05:17 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
plawless, model_based.shtml, Sat Mar  7 13:05:23 1998, 136.206.25.86, Mozilla/4.03 [en] (WinNT; I)
mollenn, summary.shtml, Tue Mar 10 15:42:16 1998, mengnt11.eeng.dcu.ie, Mozilla/4.03 [en] (WinNT; I)
mollenn, types.shtml, Tue Mar 10 15:42:31 1998, mengnt11.eeng.dcu.ie, Mozilla/4.03 [en] (WinNT; I)
mollenn, summary.shtml, Tue Mar 10 19:07:17 1998, mengnt11.eeng.dcu.ie, Mozilla/4.03 [en] (WinNT; I)
plawless, summary.shtml, Wed Mar 11 10:07:26 1998, 136.206.25.89, Mozilla/4.03 [en] (WinNT; I)
plawless, toolbox.shtml, Wed Mar 11 10:07:30 1998, 136.206.25.89, Mozilla/4.03 [en] (WinNT; I)
mansukb, summary.shtml, Wed Mar 11 20:09:18 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mansukb, summary.shtml, Wed Mar 11 20:10:45 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mansukb, boi.shtml, Wed Mar 11 20:10:57 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mansukb, boi.shtml, Wed Mar 11 20:10:57 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mansukb, history.shtml, Wed Mar 11 20:11:11 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mansukb, types.shtml, Wed Mar 11 20:11:19 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mansukb, perspec.shtml, Wed Mar 11 20:11:30 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mansukb, learning.shtml, Wed Mar 11 20:11:37 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mansukb, procon.shtml, Wed Mar 11 20:11:45 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mansukb, applic.shtml, Wed Mar 11 20:11:53 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mansukb, toolbox.shtml, Wed Mar 11 20:12:02 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mansukb, example_2.shtml, Wed Mar 11 20:12:12 1998, duff, Mozilla/3.0Gold (X11; I; SunOS 5.5.1 sun4u)
mollenn, summary.shtml, Thu Mar 12 11:48:24 1998, mengnt11.eeng.dcu.ie, Mozilla/4.03 [en] (WinNT; I)
khadirmt, summary.shtml, Sat Mar 21 17:24:52 1998, mengnt12.eeng.dcu.ie, Mozilla/4.03 [en] (WinNT; I)
khadirmt, boi.shtml, Sat Mar 21 17:26:09 1998, mengnt12.eeng.dcu.ie, Mozilla/4.03 [en] (WinNT; I)
khadirmt, history.shtml, Sat Mar 21 17:26:15 1998, mengnt12.eeng.dcu.ie, Mozilla/4.03 [en] (WinNT; I)
khadirmt, types.shtml, Sat Mar 21 17:26:17 1998, mengnt12.eeng.dcu.ie, Mozilla/4.03 [en] (WinNT; I)
khadirmt, perspec.shtml, Sat Mar 21 17:26:20 1998, mengnt12.eeng.dcu.ie, Mozilla/4.03 [en] (WinNT; I)
```