COMP9444 Neural Networks
COMP9844 Extended Neural Networks
Teaching Period 2 2013

Contents

- Course Staff
- Course Details
- Course Aims
- Student Learning Outcomes
- Assumed Knowledge
- Teaching Rationale
- Teaching Strategies
- Assessment
- Academic Honesty and Plagiarism
- Course Schedule
- Resources for Students
- Course Evaluation and Development
- Other Matters

If you are looking at a printout of this web document, and want to access the links, or if you want a more printer-friendly version, go to http://www.cse.unsw.edu.au/~cs9444/intro2013s2.html

Course Staff 2013

Contact Details

Bill Wilson will not be involved in COMP94444/9844 in 2015 (if it runs).

Staff member	A/Prof. Bill Wilson Coordinator	Dr Anthony Knittel Lecturer
E-mail	billw *	aek *
Phone	9385 6876	
Office	K17-405	
Appearance		

* @cse.unsw.edu.au

Consultations

	Bill Wilson	Anthony Knittel
Consults	K17-405 2pm Mondays	K17-402 2pm Wednesdays please email in advance
Pre-exam Consultations	K17-405 10am-12.30pm Tue 19 & Wed 20 Nov and/or email your questions (about Bill's lectures) to billw	K17-402 2pm-4pm Mon 18 Nov & Tue 19 Nov and/or email your questions (about Anthony's lectures) to aek

Each lecturer runs consultations only in weeks when he is lecturing or when one of his assignments is running (and there will be revision-period consultations - times to be announced when the examination timetable is known.)
If you can't make it to our scheduled consultations - e.g. because they clash with another of your classes - then please contact us by phone or email for an appointment.
Consultation times may have to be changed in certain circumstances.
It is best **not** to send e-mail queries to the cs9444 account; mailing to the relevant lecturer is likely to be more reliable, because the spam density on the cs9444 account is over 99%. *Note that other CSE courses may have different requirements for contacting staff.*

Course Details

Units of credit

COMP9444/9844 earns you 6 Units of Credit. This signifies 25% of a full-time study load for one semester, and involves 12 weeks × 3 hours/week of lectures (4 hours/week for COMP9844), plus 4 assignments.

Parallel Teaching Information

This course allows both senior undergraduates and postgraduates to enrol. Only COMP9444/9844 students attend these classes.
COMP9444 is regular Neural Networks, and has both PGs and UGs in it.
COMP9844 is Extended Neural Networks, and has both PGs and UGs in it.

Handbook Entry

The handbook entry for COMP9444 can be found here.
The handbook entry for COMP9844 can be found here.

COMP9844 Extended Neural Networks

COMP9844 Extended Neural Networks is also a 6 Unit of Credit Course, and so it constitutes 25% of a full-time study load for one semester. COMP9844 provides extra challenges for those who wish them. COMP9844 students attend COMP9444 lectures, and also a "bonus" lecture that covers the extension material. The extensions may include extra components to assignments.

Extended courses are intended for students with a mark of at least 70 in the prerequisite course. COMP9444/9844 has a complicated prerequisite structure, partly to deal with the fact that some students may take it in their first semester of study at UNSW. Such students may choose COMP9844 but should monitor their performance in their first 6 weeks of semester and consider carefully whether to switch to COMP9444 towards the end of this period. If you have taken one of the prerequisite courses (e.g. COMP9024, COMP2011, COMP2911) then you should not enrol in COMP9844 unless you scored at least 70 in that course.

Because COMP9844 and COMP9444 are closely related, it is possible to transfer between them past the **week 1 deadline for adding a course**. However, it is still necessary to have your final enrolment sorted out by the HECS Census Date (31 August in second session).

Extended Neural Networks will cover the syllabus of COMP9444 together with additional topics, to be presented in the additional lecture hour. See the lecture schedule for the planned content. Information on assessment in COMP9844 is below.

In years when the the enrolment in COMP9844 is very small (1-3), the extra lecture may be replaced by a reading course. (This looks unlikely for 2013s2, as there are currently 14 students enrolled [11 July 2013]). If the lecture were replaced, lecture notes would be available on the web as usual, and COMP9844 students would be able to come and discuss the lecture material in the lecture slot on Tuesday afternoon, or perhaps another time by arrangement.

Lectures

The lecture times are:

Session 2, 2013	Day	Time	Location	Weeks
COMP9444/9844	Monday	18:00-21:00	CLB-3	1-9, 11-13*
COMP9844 only	Tuesday	14:00-15:00	CLB-2	1-9, 11-13*

*No Monday lecture in week 10 because of October long weekend public holiday.
No Tuesday (COMP9844) lecture in week 10 for the sake of consistency.

Lecture Attendance

If you do not attend a lecture, then it is your responsibility to discover any announcements and/or supplemental information that's given. Lectures introduce you to the concepts covered by this subject, provide motivating examples to help you understand them, and demonstrate skills and processes. You may not understand the material completely simply by listening in lectures. You should also:

- review the lecture material after the lecture;
- discuss the material with fellow students if possible;
- read up on the topics covered in each lecture;
- complete relevant assignments and exercises, if any;
- consider exploring the topic on-line by writing and running your own programs;
- attend a consultation session (or phone or email) if you still don't understand the material

Course Aims

This course aims to introduce students to a range of topics in the field of artificial neural networks, and to provide them with hands-on familiarity with three of these areas.

Topics are chosen from: Network architectures: perceptrons, Hopfield and Kohonen nets, ART models, back-propagation trained feed-forward networks, recurrent nets, weightless nets. Computational complexity analysis of training neural network architectures. Probabilistic analysis of generalisation capabilities of feed-forward networks. Hardware based neural nets. Introduction to fuzzy logic, neurotrained nets; designing successful applications of neural networks; tensor product networks; and recent developments in neural networks.

Student Learning Outcomes

The numbers in [] below refer to UNSW Graduate Attributes which are addressed by the item in question. In addition to the graduate attributes indicated below, the experience of working in this class addresses graduate attributes 5, 7, 8, and 9. See http://www.cse.unsw.edu.au/~billw/gradatts.html for explanation of numbered graduate attributes.

After completing COMP9444, students should

- understand aspects of the social, intellectual, and neurobiological context of neural networks research; [2, 10, 11]
- have an understanding of a variety of neural networks techniques, including: [3]
  - error backpropagation learning
  - support vector machines
  - committee machines
  - recurrent networks
  - tensor product networks
  - self-organising maps
  - Hopfield networks
  - Deep learning networks
- be able to analyse a problem for NN solution in terms of these methods; [3]
- have an awareness of the computational theory underlying NNs; [3]
- have a working knowledge of a typical neural network simulation package: tlearn, and be able to use it to perform a range of computational tasks; [4, 6, 12]
- have experience in programming NN applications from scratch. [6]
- exposure to research techniques in neural networks and cognitive science: some topics will be based on research papers and monographs, to which references will be given in the course notes

Students will complete 3-4 programming-style assignments. [3, 4, 6]

Extra learning outcomes for COMP9844

Have a deeper and broader understanding of neural network techniques, derived from learning the material described in the extra lectures (see course schedule).

Assumed Knowledge

COMP9444/9844 can be taken as part of any of the postgraduate coursework degrees, any research degree, and any undergraduate degree in the School of Computer Science and Engineering. Students from related disciplines like Electrical Engineering & Telecommunications may also enrol with permission from their School. For undergraduates, COMP9444/9844 would normally be a final year elective, or occasionally a third-year elective.

Pre-requisite:

For undergraduates: (Data Structures and Algorithms *or* COMP2011 Data Organisation *or* COMP2911 Engineering Design in Computing 2) **and**
12 UoC of COMP3### or COMP4###

For postgraduates: (Enrolment in a postgraduate program in the School of CSE or the School of EET *or* COMP9024) **and**
12 UoC of COMP9### or ELEC9### or TELE9###

What we want you to have taken away from the prerequisite courses is a general knowledge of programming techniques, and experience with using them in other than the exercises of an introductory programming course. Also, specifically you will need to program in C, in the first assignment, and in C or some other language in the final assignment.

Mathematical assumed knowledge: We also assume you have completed basic university mathematics courses, such as the UNSW courses MATH1131 and MATH1231, and so know about concepts of vectors and matrices (including inner products and orthogonality). It is possible to learn (or revise) these concepts as you do COMP9444; obviously this will take more time than if you already know them.

Teaching Rationale

See aspects of the section titled Resources for Students.

Teaching Strategies

1. **Lectures** provide a detailed description of the material covered by the course, in an environment that provides a

chance to ask questions and listen to the answers to other students' questions. Because the lecturer can see the students' reactions, the lecturer can gauge their level of understanding, and expand on technical material that seems to need further explanation;
2. **Textbooks and reference books** provide supplementary material, and/or a slightly different point of view on the technical material;
3. **Exercise sets** allow you to test and extend your understanding of the lecture materials by tackling problems based on those materials. Some of the questions in the exercise sets might be similar to questions later used in the final examination;
4. **Quizzes** provide feedback on your progress. They are voluntary, and self-administered, at a time convenient to you.
5. **Programming assignments** provide: a chance to test and strengthen your understanding of some of the lecture material by coding it; If you can code a NN algorithm from scratch, then you almost certainly understand it well.
6. The **sample examination** gives students a chance to test themselves on the material of the course in a format similar to that of the final examination;
7. **Consultation sessions** give students an opportunity to ask the lecturers, one-on-one, specific questions about any of the course material that they might be uncertain about;
8. **Other students** - your classmates - are there to help you, too. Because they have just gone through the same learning processes as you, on the material of COMP9444, they are in a way best placed to know what the difficulties are, and explain them. By explaining them, they too will benefit, by reinforcing their own learning. The only limit on this is in relation to the assignments: you can get ideas for the assignments from other students, but you must write the assignment yourself (see plagiarism).
9. The **final examination**: nobody we know enjoys examinations (either sitting them or marking them) but they do provide a focus and a deadline to make sure you revise the material covered in the course. Revision can be an important part of learning.

Assessment

The assessment for this subject will consist of:

- 2 hour final exam worth 70%
- 3-4 programming style assignments, together worth 30% (see below)

These marks will be combined using a harmonic mean formula - see below for details - **not** by adding the two marks together.

COMP9844 students: must attempt all 4 assignments. COMP9844 students who attempt 3 assignments or fewer are penalised by subtracting 30% from their assignment total. In addition, COMP9844 students are responsible for knowing the contents of lectures in the extra COMP9844 lectures, and may be examined on all topics.
COMP9444 students: must attempt at least 3 assignments. COMP9444 students who attempt 2 assignments or fewer are penalised by subtracting 30% from their assignment total.

The assignment total is based on the sum of the marks for your *best three* assignments.

The assignment total is then scaled to give a mark out of 100, called `asstotal`. The final exam total is also scaled to give a mark out of 100, called `exam`. The assignment total and the final exam total will be combined using a **harmonic mean formula**, weighting the final exam as 70% (0.7) and the assignment total as 30% (0.3):

```
final mark = min(100, Δ + asstotal*exam/(asstotal*0.7 + exam*0.3))
```

Here `asstotal*exam/(asstotal*0.7 + exam*0.3)` is the weighted harmonic mean of `asstotal` and `exam`, and Δ is the average difference, across the whole class, between the weighted arithmetic mean and the weighted harmonic mean of `asstotal` and `exam`. (The weighted arithmetic mean is 0.7*`exam` + 0.3*`asstotal`. The weighted harmonic mean is always less than or equal to the weighted arithmetic mean, so $\Delta \geq 0$).

The weighted harmonic mean penalises uneven performances: getting 100% (30/30) for the assignments and 30% (21/70) for the exam would allow a pass if marks were simply added together, (i.e. if the weighted arithmetic mean were used) but under the harmonic mean formula, it results in a final mark of 38%. The effect of the Δ term is to ensure that, on average, the class is not penalised by the use of the harmonic mean. Those with very different assignment totals and exam totals will still do worse, but those with moderate differences will not do worse, and those with even performances will do better than if a weighted arithmetic mean had been used.

More on harmonic means.

Criteria by which marks will be assigned

These are explained in each assignment specification.

Procedure for submitting assignments

This is explained in each assignment specification. Some assignments are handed in at the lecture in the week in which they are due, and some are submitted electronically, or partly electronically.

Policy on Late Submission

Late submission of assignments: Assignments submitted late are subject to the following penalty: the maximum mark obtainable reduces by 10% per day late, up to a maximum of 50%. Thus if the assignment is marked out of 10, and students A and B hand in assignments worth 9 and 7, both two days late, then the maximum mark obtainable is 8, so A gets min(9, 8) = 8 and B gets min(7,8) = 7. Assignments handed in over a month late, or after (if) a solution has been released, will normally receive no marks.

Rationale for Each Assignment

See section on Relationship between objectives and assessment.

Warning about High Assignment Marks

Be aware that assignment marks in COMP9444/9844 are likely to be high. In the past, some students have thought that this means that they do not need to study much for the exam. This is *not* wise. Take a look at the 2007s2 mark distribution statistics. You will see that, for example, while at least 68% of students scored in the HD range in each assignment, only 21% scored in the HD range in the exam (i.e. exam mark ≥ 85).

We (the lecturers) understand that this may make the final exam rather stressful, and we have considered other assessment models. However, we take the view that end of semester is the best time for most of the assessment weighting, as by then you have had as long as possible to understand, and revise, the course material.

The self-administered feedback quizzes (where available) allow you to monitor your progress: they don't count towards the assessment, but they do allow you to decide whether you understand the lecture material.

Assessment for COMP9844

COMP9844 assessment will have the same structure as for COMP9444 (3 hour exam worth 70% + assignments worth 10% each, as described above). However, assessment components for COMP9844 may (probably will) differ from those for COMP9444. For example, the COMP9844 assessment may have extra components compared with the corresponding COMP9444 assessment item. There will be enough in common between the assessment items for COMP9444 and those for COMP9844 that it will be possible to mark both on a consistent and equitable scale.

Students enrolled in COMP9844 must make a reasonable attempt at any extra assignment material, and must attend 80% of the extension lectures, or present (a) medical certificate(s), or they will be denied assessment in COMP9844. This rule will be administered in a humane way, and is intended to stop people from enrolling in COMP9844, but doing none of the extension work, thus ending with a grade in "Extended Neural Networks" rather than "Neural Networks" without any extra effort.

Relationship between objectives and assessment

The programming assignments provide a practical introduction to Neural Network programming using a number of types of neural networks.

The final exam tests students on the learning objectives that are not covered in the assignments (see above for list). However, in practice the assignments have a formative role in assessment as well: we have observed that students who do not complete the assignments satisfactorily normally fail the exam.

More details are in the table below.

Task	Learning Outcomes
Backprop assignment	understanding of the backprop learning algorithm
SVM assignment	understanding of support vector machines
Kohonen assignment	understanding of the self-organising map algorithm
Recurrent nets assignment	understanding of simple-recurrent nets
Final exam	tests the following topics: multi-layer perceptrons learning theory support vector machines committee machines recurrent networks self-organising maps and LVQ algorithm tensor product networks Hopfield networks and BSB algorithm see also sample exam questions, linked at week 12 of Course Schedule

For COMP9844 students, the exam also tests the topics covered in the extra COMP9844 lectures.

Assignments

Assignments give you the opportunity to practice what you have learnt. Assignments are a very important part of this subject, therefore it is essential that you attempt them yourself. There will be four assignments, on:

1. Error Backpropagation Learning ("backprop")

Download the MLP_numbers.tgz tar archive file containing software as well as training and test data. The first assignment provides a practical introduction to neural network programming using the backpropagation paradigm and the C programming language.

2. Recurrent Networks
   The third assignment allows students to test their understanding of simple recurrent networks, as applied in this case to a toy problem in financial prediction. The tlearn neural network simulation package is used. Students write a report on their experience with tlearn, as well as handing in their tlearn files.
3. Support Vector Machines
   Support Vector Machines
   The third assignment provides a hands-on approach to finding the maximum margin classifier for a support vector machine, and the use of kernel methods.
4. Kohonen Networks
   The final assignment gives students a chance to code the SOM algorithm from scratch, and experiment with applying it to learning about clusters in data. Students may choose the programming language that they use in this assignment, within reason. In the past, students have used C, C++, Java, python, and matlab.

When available, the assignment specifications will be linked above.

Assignments are to be completed in your own time. You must submit your assignments for marking via the Unix `give` command and/or by handing in a print-out at a lecture (more details will be made available to you as part of the assignment specification). To maximise the learning benefits from doing assignments, it is essential that you start work on assignments early. Do not leave your assignments to the last minute.

Academic Honesty and Plagiarism

Plagiarism is defined as "using the words or ideas of others and presenting them as your own". UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Learning Centre: Plagiarism and Academic Integrity
- MyUNSW: Plagiarism and Academic Misconduct
- CSE: Addendum to UNSW Plagiarism Guidelines
- CSE: Yellow Form (whose terms you have agreed to)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism.

Course Schedule

Now updated for 2013 semester 2

The order and content shown below may vary - what is shown below is an indication only.

ecture notes may be updated a few days before the lecture.

COMP9444 Lectures (2013s2 dates)						
Wk	**Date**	**Lecturer**	**Topic**	**Notes**	**Remarks/Planned assignment dates**	**Feedback Quiz**
1	29 Jul	Anthony	Introduction and brain physiology	1x1.pdf 2x2.pdf		
2	5 Aug	Anthony	Perceptrons and Backpropagation	1x1.pdf 2x2.pdf	Assignment 1 out	
3	12 Aug	Bill	Recurrent networks 1	pdf (2.2Mb)		quiz
4	19 Aug	Bill	Recurrent networks 2	As for week 3		quiz
5	26 Aug	Anthony	Applications, Deep Learning Networks	1x1.pdf 2x2.pdf	Assignment 1 due Recurrent networks assignment out	
6	2 Sep	Anthony	VC-dimension, PAC-learning and Structural Risk Minimisation	1x1.pdf 2x2.pdf		
7	9 Sep	Anthony	Support Vector Machines	1x1.pdf 2x2.pdf		
8	16 Sep	Bill	Self-organising Systems	pdf (2.3Mb)	Recurrent networks assignment due Assignment 3 out	quiz
9	23 Sep	Bill	Neurodynamics - Hopfield nets Brain-State-in-a-Box (BSB)	pdf (1.3Mb)		quiz
	30 Sep-4 Oct: Mid-session Break					
					Assignment 3 due	

10	7 Oct	No lecture (Labour Day public holiday)			Kohonen networks assignment out	
11	14 Oct	Anthony	Committee Machines, Recent Applications	1x1.pdf 2x2.pdf		
12	21 Oct	Bill	Tensor-product Networks	pdf (772 Kb)		quiz
13	28 Oct	Both	Revision: Sample exam	Questions Part A Solutions Part A Questions Part B Solutions B	Kohonen networks assignment due on Friday	

Extra COMP9844 Lectures						
Week	**Date**	**Lecturer**	**Tentative Topic**	**Notes**	**Remarks**	**Quizzes**
1	30 Jul		No Tuesday lecture in week 1			
2	6 Aug	Anthony	Autoencoders I	1x1.pdf 2x2.pdf sample code		
7	13 Aug	Bill	Dynamic nets	pdf notes (1.9Mb) Fahlman paper (160Kb)		quiz
8	20 Aug	Bill	Radial basis function nets	pdf (1.2Mb)		quiz
3	27 Aug	Anthony	Autoencoders II	1x1.pdf 2x2.pdf sample code		
4	3 Sep	Anthony	Pretrained MLPs	sample code		
5	10 Sep	Anthony	Restricted Boltzmann Machines	sample code		
9	17 Sep	Bill	Competitive learning	pdf (970Kb)		quiz
11	24 Sep	Bill	Simulated annealing	pdf (355Kb)		quiz
	30 Sep-4 Oct	Mid-session Break				
10	8 Oct		lecture cancelled - remaining lectures will be in weeks 11-13 rather than 10-12			
6	15 Oct	Anthony	Deep Belief Networks	notes		
12	22 Oct	Bill	Boltzmann machines Systematicity	pdf pdf		quiz quiz
13	29 Oct	Bill + Anthony	Revision - Sample exam questions	Questions - pdf (55K) + Solutions - pdf (170K)		

Resources for Students

Course Resource List

Details are below for many of these - follow the links:

- Text Book. The textbook allows you to expand or clarify your understanding of lectures. While you are of course welcome to read the textbook ahead of the lectures, this is not assumed.
- Lecture Notes and Lectures. The lectures provide a guide to, and exposition of, the course material. The lecture notes provide a fairly comprehensive summary of what is said in the lectures, additional comments may be made in lectures that are not covered in the lecture notes, *and you are responsible for finding out about anything extra that was said in lectures, if you happen to miss the lecture.*
  At least some of the lectures should be available on Lectopia after they have been delivered.
- Revision Notes on Linear Algebra
- Self-feedback Quizzes. May not be available for every week. These allow you to tell how well you have understood the the material covered in the week which they address.
- Exercise Sets. These expand on and give extra examples of the material covered in lectures.
- List of relevant journals & conferences
- tlearn software. The `tlearn` package is used in one of the assignments.
- your fellow students - get to know them and talk to them - because they, like you, are committed to learning about Neural Networks, they can help you to learn, and, by doing so, they will help clarify concepts for themselves. Just remember to stay on the right side of the plagiarism rules, when discussing assignments.
- the COMP9444/9844 class forum - which potentially has contributions from both staff and students
- lecturer consultation sessions
- the UNSW Library has other Neural Networks resources - go and find them!
- and of course there's the web

This is a technical course, and you will be expected to understand the technical aspects, including remembering some of the formulae, as well as explaining their significance, and to be able to apply the technical details to creating simulated neural networks and solve problems using theoretical concepts covered in the course.

Self-Feedback Quizzes

See the links labelled "quiz" at the end of rows in the course schedule, or click on "Quizzes" in the green menu on the class home page, or go here. These quizzes are voluntary self-assessments. They carry no marks. They are provided to help you work out how well you are going in COMP9444/9844.

Solutions are linked to each quiz.

We suggest that, around the end of each week, you set aside some time to review the COMP9444 lecture material for the week, then try the quiz. Then look at the answers, and mark yourself. If you do not score as well as you aimed to score, then obviously you need to revise more extensively.

You can also get feedback from assignments (which are assessable), and from exercises - see below.

Exercises

These are more problem-oriented, and involve longer answers than the quizzes. Again, they are not assessable. You can most easily find them in the menu at the left of the class home page, but here is another link to the exercises. Solutions are provided to exercise sets.

Text Book

Haykin, S., *Neural Networks and Learning Machines*, Prentice-Hall, 3rd ed., 2009.
The second edition had a slightly different title: Haykin, S., *Neural Networks: A comprehensive foundation*, Prentice-Hall, 2nd ed., 1999. If you already have access to the second edition, you should be able to manage with that. If you are buying now, we suggest getting the 3rd edition.

Recommended Reference Books

On the role of neural networks and connectionism as framework for developing AI systems:

Hoffmann, A.G. Paradigms of Artificial Intelligence - A methodological and computational analysis, Springer-Verlag, 1998.

Further Links and References

ALVINN Autonomous Land Vehicle in a Neural Network

TD-Gammon Neural Network Backgammon Player

Tutorial on Support Vector Machines, by Christopher Burges

Wikipedia AdaBoost

K.J. Lang and M.J. Witbrock, 1988. *Learning to tell two spirals apart*, in Proceedings of the 1988 Connectionist Summer School, published by Morgan Kaufman.

Relevant Journals and Conferences

Here are the names of some of the more prominent Neural Network journals and conferences:

- J: Neural Networks
- J: IEEE Transactions on Neural Networks
- J: Neural Computation
- J: Connection Science
- C: NIPS (Neural Information Processing Systems)
- C: ICONIP (International Conference On Neural Information Processing
- C: for conferences organised by the International Neural Network Society, see here, under Announcements.

Some material on neural networks appears in more general journals and conferences. Places to look for the cognitive science end of NNs include:

- J: Behavioral and Brain Sciences
- J: Artificial Intelligence
- J: Cognitive Science

Software Availability

tlearn

- tlearn user manual (pdf, about 40 pages, 719KB) and Unix manual for tlearn (Postscript, 7 pages, 47K);
- Center for Research in Language Resources page, from which you can download a Windows GUI version of tlearn,

as well as source code for a Unix version (but see next dot-point). If you can't find the *source* to xtlearn here, try Bill's squirreled away copy of the xtlearn source. If that doesn't work, try http://www.cse.unsw.edu.au/~billw/xtlearn.tar.

- xtlearn is available for CSE's pc.i86.linux machines, at `~cs9444/bin/xtlearn`). Note that the interface to `xtlearn` is a bit different from the Mac interface, on which the documentation is based. The Mac version has a built-in text editor, whereas in the Unix (X-windows) version, you need to use your own favourite text editor to create the `.cf`, `.data`, `.teach`, etc. files. Students have complained that it freezes under certain circumstances if the "Num Lock" key is on (pressed). The fix is: press the "Num Lock" key again to turn it off `:-)`.
- A command-line version of tlearn is also available for pc.i86.linux machines. Unfortunately, it doesn't seem to work correctly. It might have been useful for batch-style simulations - e.g. where you wanted to run, say, several variations on a simulations of a particular model from a script, and then, maybe, automatically analyse the output. The Unix manual for tlearn (linked above) describes this variant. We mention it because while batch tlearn itself doesn't seem to work, you can still get some of the benefit, because xtlearn understands the switches to the command line interface. Thus you can write a script to, e.g., specify all the training options as command line options. Then all you have to do is sit there and click "Train network" and "Quit" each time xtlearn's GUI opens. Thanks to Johan Bruun for pointing out the problems with tlearn.
- Info on Cross-Entropy as an error measure in `tlearn`. See also the Training Options menu item in `tlearn`.
- Naturally, if you use a version of tlearn different from the standard CSE Linux one to develop an assignment, remember to do your final run on a CSE Linux machine, using the standard CSE Linux version of tlearn.

Theano (for 9844)

- The Theano module for python can be installed on a CSE machine, by running:

  ```
  easy_install --user theano
  ```

  Commands can be entered at the python prompt. It may not be practical to run long experiments on CSE machines, and the built-in linear algebra libraries may be slow (it complains about failing to import scipy.linalg.blas), however it should be sufficient for running commands to become familiar with them.
- Demonstration code is available in this folder
- Further information about installing Theano locally is available here, as well as a tutorial on Theano commands here. Extensive knowledge of Theano commands is not expected for the course, it is being used as a tool for teaching the manner in which neural network designs are implemented.
- Tutorials on implementations of Multi-layer Perceptrons, Convolutional networks, Autoencoders, Stacked autoencoders, RBMs and Deep Belief Networks are available here

Matlab

- Haykin's Matlab code for the second edition of his book.

Tensor Product Networks

- C code for Tensor Product Net operations (simplified version of STAR-1 system)

Course Evaluation and Development

COMP9444 is evaluated by survey each time it is taught. The survey includes stock questions asked of all comparable courses, so that it is possible to compare the delivery of COMP9444 with other relevant UNSW courses, and also a space for free-format comments. You cannot be identified from your survey response, unless you deliberately include identifying information in your free-format comments.

The completed survey forms are analysed statistically by "CATEI" software. CATEI is run by UNSW Central according to a timetable controlled by UNSW Central, not the School of CSE and certainly not the staff of COMP9444/9844.

The comments and statistics reported are used to improve the course, and *all* UNSW lecturers are required to summarise the survey results for their Head of School, and report on what improvements they plan for their course when it is run next.

In the surveys conducted in recent sessions, the summary criterion "Overall, I was satisfied with the quality of this course" received the following responses:

Session	Strongly Agree	Agree	Mildly Agree	Mildly Disagree	Disagree	Strongly Disagree	Mean Rating	School Average	Response Rate
2008 s2	27%	73%	0%	0%	0%	0%	5.27	4.74	16/28 = 57%
2009 s2	46%	35%	15%	4%	0%	0%	5.23	4.75	26/41 = 63%
2010 s2	50%	0%	50%	0%	0%	0%	5.00	4.64	4/13 = 31%
2011 s2	25%	67%	8%	0%	0%	0%	5.17	4.62	12/30 = 40%
2013 s2	41%	38%	13%	6%	3%	0%*	5.16	4.52	32/66 = 48%

* Percentages may not add to 100 due to rounding

In looking at the free-format comments, no strong patterns emerged that indicated a need for changes:

- four students commented positively on the assignments, and one negatively
- three other students liked the "balance" of the course - between depth and breadth, or between coding and theoretical work.

Comments on COMP9444 and COMP9844 included:

- "Best features of this course were course content, course materials and the support provided to students by the lecturers."
- "It is nice to understand how simple and powerful a neural network is. And the assignment really help to understand how the network works and how it preforms."
- "Challenging concepts and interesting materials."
- "Great and broad coverage of the material, attention to detail in the mathematical content."
- "Handling real neural networks and developing learning systems to solve real-world problems."
- "The fact that the course generalizes from basic concepts very fast. Also good was that the course material is very linked to relatively recent papers, and teaches cutting-edge research on the topic."
- "Being able to get a broad higher level understanding of what makes neural networks and learning was very interesting from the earlier lectures. The assignments were also great at encouraging students to understand the topics. The online exercises were also useful."
- "good balance between practical and theoretical assignments interesting subject content"

Two students commented on timetabling issues that are unfortunately outside of the lecturers' control.

Other Matters

- Students enrolled in COMP9444 and COMP9844 are expected to attend all classes
- The use of School of Computer Science and Engineering computing laboratories is subject to rules described in the Yellow Form, which you acknowledge (electronic) receipt of when you receive your computing account. The Yellow Form also outlines what to do in case of illness or misadventure that affects your assessment, and supplementary examinations procedures within the School of Computer Science and Engineering.
- Occupational Health and Safety, and Ergnomics (safe use of computers). To find out about UNSW policies in these areas, go to the Yellow form section on OH&S, which has links to the latest edition of these policies. The most immediate area of concern for CSE students is ergonomics and repetitive strain injury. See the CSE OHS & Ergonomics page.
- UNSW Equity and Diversity Policy Statement
  UNSW Student Equity and Disabilities Unit