Formal Computational Skills

Projects

Today's lecture:

- The rationale behind the assessment

- How it will be assessed

- 3 Example topics in detail

    - Analysis of a neural network
    - Optimisation
    - Analysis of a dynamic neural model (continuous time recurrent neuron, GasNet neuron, spiking model etc

- Other example topics

Main Idea

Project is to investigate a mathematical subject that you want to use, or analogously, investigate a subject you want to use mathematically.

Rationale is that, when you come to use it, you will have an intuitive understanding of how it works

You should pick a subject you do not understand now and try to understand it via analysis, simulation, experimentation etc. Basically: pulling it apart and putting it back together again

It is a good opportunity to really understand a topic and to gain an intuitive feel for a bit of a course you are interested in

As such, it is probably best to start with a toy example eg if you are using a network, start with a simple one (see egs later)

Choosing a topic

Topics you choose must be agreed with me so that they are:

a) not too easy
b) not too hard
c) appropriate for the course

To this end, if you are going to do a different topic to those described subsequently, you will need to discuss it with me (via email, in seminars etc)

However, the 3 examples given today will be subjects covered in the subsequent 3 weeks as I DO NOT expect you to pick a topic now

Usually, people start once the coursework is starting to end.

An example

I’m interested in GasNet type model neurons for robot control:

I could evolve a network to control a robot but what would this tell me? Probably little – its hard to get evolution to work especially if I don’t really know how GasNet neurons function

Much better to:

a) look at a single GasNet neuron, see what parameters affect its behaviour, get a feel for what happens given different inputs

b) Look at evolutionary algorithms: its known that robotics spaces have a lot of neutrality and noisy fitness evaluations. Make a space with KNOWN characteristics and see how evolution copes with these problems

c) Evolve 1 neuron for a simple task:

All 3 allow thorough analysis unlike the 1st scenario

Assessment

Assessment is not based on mathematical complexity, but rather on demonstrating the process of investigation and showing that you have understood the topic at hand.

A good way of doing this is to imagine that you are explaining your subject and findings to someone else (me in this case)

The level of detail/mathematics needed (ie how much you explain about the subject) is therefore dependent on what YOU need to understand/have understood

In report, show me what helped you gain an understanding and explain the techniques you have used

Originality of ideas/explanation etc will be rewarded

Project is worth 50% of the marks for the course. However you will already have roughly 50% from doing the other coursework Since top projects for other courses are, in principle, publishable and, it is unlikely you will discover a new branch of maths, marking scheme here needs to be different:

Project looking at optimisation, ran a GA, changed the mutation rate, presented the results with no analysis/explanation <10/50

Project analysing a CTRNN. Didn't go much past analysis done in seminars. Leant heavily on a Beer paper 10-19/50

Project analysing ctrnn. Tried a few of them in tandem. Tried analytical techniques not seen in lectures. Discussed properties that made them good/bad for evolution 20-29/50

Examined fourier transforms for evolving music. Nice intro, showed what had been learnt. Good analysis of problem. 30-39/50

New analytical methods for examining neural models. 40-50/50

Other bits

You could work in groups: it means you can take on a broader selection of things. In the previous example, one could look at a GasNet neuron the other at evolution etc. Talk to me if you fancy this option.

I will give feedback on an outline of the project (ie page saying main headings etc) if handed in a few weeks prior to the deadline. Or can be used to get help on ideas

Get peer feedback on the outlines before/after/without me. May be better to get someone who isn't doing the same project as you. Good to get them to write down some comments: helps to formalise things. Or, note things you like and things that could be improved

EG 1: Analysis of a neural network

Use the network from this week's coursework as a starting point

1) Investigate what functions it can produce
2) What happens if you add noise to the training data?
3) What happens if you use non-linear activation functions like the sigmoid function or a step function (as in the perceptron: see Bishop etc)?
4) What's the point of adding a bias node?
5) What do recurrent connections buy you?
6) What about adding other layers – does it help if you don't use non-linear activation functions?

Compare what should happen theoretically with what happens in practice

As with all the examples suggested, the above are only ideas of things that you could change/investigate.

You wouldn’t have to do them all and could focus on only one

The idea is to play with the network, see what it does and see why certain things (such as biases and non-linear activation functions) are used

However, as you can control the complexity of the problem via making the target functions more or less complex, you can really get an idea of what networks are good/bad at and what various learning algorithms are good/bad at

For refs, have a look at Bishop regarding perceptrons (and multi-layer perceptrons) and other perceptron refs (eg Haykin, though it’s a bit heavy on the maths)

EG 2: Optimisation

Use the gradient ascent example (next week's task) as a starting point to investigate issues raised

1) Improve gradient descent convergence
2) Compare the performance of different algorithms
3) What happens on landscapes with different properties?
4) Investigate methods of mutation for hill-climbing and eg error thresholds
5) How does a population-based method differ?
6) What happens with a GA?
7) What happens when the function evaluation is noisy (as in robot experiments)?

Various refs: see Haykin, ch. 4 and Bishop, ch 7 etc

Do not simply optimise a difficult problem. You must be able to analyse why things happened and why things went well

EG 3: Analysis of a Neural model

Lots of dynamic neural models: neural oscillators, spiking models, ctrnns. Will use ctrnn to illustrate the types of project Neuron is governed by the following equation:

$$\tau_i \frac{dy_i}{dt} = -y_i + \sum_{j=1}^{N} w_{ji}\sigma(y_j + \theta_j) + I_i(t)$$

where y_i is the state of the neuron (output) τ_i is a time constant (normally between 1 and 10), I_i is external (sensory) input, w_{ji} is the weight of the connection from the jth to the i'th neuron, θ_j is a bias term and $\sigma(x) = 1/(1+e^{-x})$ is the exponential sigmoid

This is normally solved by Euler integration (see lecture in 2 weeks for details of numerical integration techniques), In the next 3 weeks we will build and analyse 1 and 2 neuron systems of CTRNN-like neurons

This investigation could form the basis of your project, either extending the analysis or repeating with a different neural model

1. Assuming only one neuron (ie all $w_{ij} = 0$), implement the above equation assuming constant sensory input.
2. Try different values for the sensory input, and time constants and see how they affect the state of the neuron
3. Analyse the stability of the neuron using simple dynamical systems analysis (see lecture in 2 weeks)
4. Could then look at varying the type of input it receives (ie make the sensory input vary over time
5. Try connecting 2 neurons together and again investigate stability. See what happens if you vary values for biases and weights

As for the analysis of the simple neural network, you would not do all the above, but start simple and make more complicated to further your understanding of all the elements

Focus on what helps YOU get an understanding of the workings of the neural model

Alternatively, do not focus on the neural model but rather look at the different techniques used for numerical integration

Or could focus on the analytical tools used in a dynamical systems analysis: what works well, what doesn't, is there anything else you could do?

Questions????

Beer refs: have a look and see the *type* of investigation performed

Beer, R.D., Chiel, H.J. and Gallagher, J.C. (1999). Evolution and analysis of model CPGs for walking II. General principles and individual variability. J. Computational Neuroscience 7(2):119-147.

Chiel, H.J., Beer, R.D. and Gallagher, J.C. (1999). Evolution and analysis of model CPGs for walking I. Dynamical modules. J. Computational Neuroscience 7:(2):99-118.

Beer, R.D. (1995). On the dynamics of small continuous-time recurrent neural networks. Adaptive Behavior 3(4):471-511.

See: http://vorlon.ces.cwru.edu/~beer/pubs.html

General Examples

Refs:
Haykin = Haykin, S. Neural Networks.
Bishop = Bishop, C. Neural networks for pattern recognition
Dayan = Dayan and Abbott, Theoretical Neuroscience
Numerical Recipes = Press et al., Numerical Recipes in C etc

Solving linear equations via matrix inversion techniques (pseudoinverse etc): what does the pseudoinverse compute? What happens in over/under-determined cases? Bishop ch 3 and Numerical recipes

Principal Component Analysis (PCA). How does it work? Why does it work? When does it fail in theory? When in reality? Bishop or Haykin

The whitening transformation and covariance matrices. What are covariance matrices? Why are they important? When should they work? When do they break down in practical situations? Bishop or Haykin

Analysis of Hebbian learning rules: what do they do? How do they work? What solutions do they produce? Dayan ch. 8

Information Theory and its use in Alife (Info theory and independent componant analysis (ICA), mutual information as complexity etc). Haykin and me for refs

Fourier transformation and analysis. Numerical Recipes and loads of maths books

Various neural models. Loads of computational neuroscience ones in Dayan and Neural Oscillators (Matt Williamson for refs: see me)

Techniques for estimating probability density functions (Bishop, ch 2 and 3)

Lagrangian optimisation, complexity in networks, neural coding in ants and others etc etc etc

Just make sure it is something you are interested in finding out about

However, as mentioned earlier, you don't need to decide on your topic yet

Hand in: last week of term, Thursday week 10, 4pm

Length: difficult to say as it depends how many figures/graphs and the subject etc but probably between 10-15 pages.

It's worth as much as all the coursework so should take the same time/effort

Could be pretty much done in last few weeks of term as other coursework will be finished

Any code used to produce the assignment must be included as an appendix

You need a cover sheet. Get one from postgraduate office and hand work in to postgraduate office