Examiners' commentary 2017–2018

CO3311 Neural networks – Zone A

General remarks

As usual with this course the examination covered all the major topics in the syllabus and allowed candidates to demonstrate their understanding of the applications and the limitations of the variety of networks studied as well as their ability to perform the necessary calculations to train or develop a specific example.

Question 1 proved to be the most popular as well as the most rewarding (in terms of marks gained) on this paper. A little less popular but almost as rewarding was Question 4 . Question 2 was more popular than Question 4 but, perhaps because of its part (c), turned out to be the least well answered of all the questions on this paper.

Comments on specific questions

Question 1

This question tested candidates' general knowledge of neural artificial network units, activation functions and learning rules.

For part (a), candidates were asked to compare and contrast biological neurons with the artificial ones studied during the course. Clearly there should be both similarities and differences mentioned for good answers. Illustrative diagrams were asked for, and answers without diagrams were severely disadvantaged in the information that they could convey quickly and concisely. Although there were many good answers, there were some poor ones that failed to capture the important differences between natural and artificial neurons.

Having explained the unit types, candidates were then required, in part (b) to define the terms used when working with them. Definitions of ***unit***, ***weight***, ***activation*** and ***net*** were required. On the whole candidates had little difficulty achieving good marks for these.

A major distinguishing feature of the different types of unit is their activation functions. Candidates were asked to list three of these and give a formula for the activation and the types of network in which they are found. On the whole part (c), was well answered.

The final part of the question, part (d) required candidates to compare and contrast Hebb's Rule and the Widrow-Hoff rule. Although there were some very good answers, more generally this part of the question was not very well answered. Good answers gave a definition of each rule followed by similarities and differences between them. Amongst relevant points were types of networks in which each is used, application areas and advantages and disadvantages of each in particular circumstances. Overall this part of the question was very poorly answered.

Question 2

This question tested candidates' understanding of Perceptrons, both the theory and the practice. Given that these are the simplest 'networks', on the whole answers were disappointing.

A Perceptron unit can be thought of as dividing the plane into two parts, and part (a) asked candidates to explain this. Good answers gave a clearly labelled diagram showing how the plane is divided into two parts, one where the Perceptron output is zero and one where it is one. Mention that the line itself belongs to the 'activation = 1' case was sometimes omitted -- this is a very important point that should be made.

Candidates were expected to show the correspondence between the weights of a Perceptron and the usual equation of a straight line. Good answers not only explicitly gave the mapping between weights and line parameters but also mentioned that, unlike the equation given, the case 'x = constant' can also be catered for using a Perceptron. The question required a diagram, but some candidates omitted this and thus lost what should have been straightforward marks.

For part (b) candidates were asked to explain the process that can be used to build a Perceptron network to model an arbitrary truth table. Good answers gave a step by step account of the algorithm, illustrating each step using XOR as example.

Weaker answers restricted themselves to this case only and omitted any reference to the general case. It was surprising that some candidates produced a Perceptron modelling the OR function omitting the exclusivity. It is only this latter constraint (exclusivity) that requires a Perceptron network rather than a single Perceptron.

For part (c) candidates were asked to design a two-input network of Perceptrons (threshold units) which produces an output of 1 if and only if both of its inputs are between 0.5 and 1.5. This proved more difficult for many candidates than it should have. If one thinks of a Perceptron that checks for a single input greater than 0.5 and one that checks for a single input being less than 1.5, then clearly putting these two together using an AND Perceptron one can check for a single input between 0.5 and 1.5. Two of these together then has the required result.

Few answers gave wholly credible schemes, but marks were awarded for reasonable strategies. Perhaps candidates were thrown by the fact that the constraints were not integral, but this is no problem for inputs to a Perceptron. This question proved to be the most difficult.

Question 3

This question tested candidates' understanding of Backpropagation networks and the Backpropagation training algorithm.

For part (a) a list of the essential features of such a network was required, as well as an illustrative labelled diagram. Some candidates chose to use a different notation than the preferred one for the course and, perhaps because of this, omitted important parts of this type of network. The course's notation is designed to capture the essential features and candidates are strongly advised to use it.

The details of the algorithms and formulae for both forward and back propagation were required in part (b). Good answers explained each term and variable in the formulae given. Good answers also gave clear explanations and used the notation introduced by the course. Poor answers were unstructured and often included confused notation. This part of the question was reasonably well answered overall.

The final part of this question, part (c) involved candidates working through a simple example. Weights, inputs and targets in the question were chosen to lighten the computational load on candidates.

This part proved straightforward for the candidates who made a serious attempt at it, but many omitted it.

Question 4

Kohonen Grossberg networks were the topic of Question 4, with aspects from: normalisation, the use of Grossberg layers and the training algorithm being tested in various parts.

Part (a) tested candidates' understanding of how the scale used for different variables might affect such networks. Few candidates were able to give a full and convincing answer to this part, despite the subject guide giving a detailed account of the issue.

Normalisation, covered in part (b), seems to have been much better understood. However, the fact that one cannot normalise the input data if examples of length zero occur in the data was overlooked by some. Good answers explained how the use of normalisation speeds up calculation in general, by reducing the dimension and enabling use of the inner product instead of length.

The Grossberg layer is often omitted from applications of these networks as more sophisticated post processing is often needed. Most answers to this part (c) were good.

The final two parts of the question (d) and (e) required the training algorithm to be stated and used. It was disappointing to see that, despite normalisation being mentioned in part (b) some candidates ignored this important aspect of the algorithm. Initial units and training data were chosen to make calculations less onerous than they would be in a general case and so candidates were given an example with 4 units and 2 training examples to train on. On the whole candidates had little difficulty obtaining good marks, though there were too many arithmetic slips lowering the average marks obtained. Poor answers tended to omit looping or other important details of the algorithm.

Question 5

Question 5 was mainly about Hopfield networks but started by asking candidates to contrast these with backpropagation.

Part (a) which asks for the main differences between the architecture and use of Backpropagation Networks and Hopfield networks was well answered and good average marks were achieved. Good answers mentioned outputs, learning rules, synchronicity and feedback.

The weights of a Hopfield network can be determined in several ways and part (b) asked how this is done in a typical application. Trial and error, Hebb's rule and Widrow Hoff and 'inspection' of the problem were amongst the techniques given by candidates. This part proved to be the hardest for the average candidate.

Parts (c) and (d) allowed candidates to display their knowledge of the working of these networks by requiring them to calculate both the state transition table and from it the state transition diagram of a simple example. As in previous years many good answers were given. Poorer answers were, in the main, spoilt by arithmetic errors or slips in the setting up of the tables. It is important that units are listed in reverse order for the states to come out in an easy fashion. This is another area where the notation designed for the subject guide helps avoid common slips. Overall high marks were gained for this part.

Some state transition diagrams were spoilt by the omission of circles showing transition from a state to itself. There were a few examples of diagrams not matching the table and it is hard to award 'follow through' marks when this is the case.

Question 6

This question concerned some of the applications of artificial neural networks. Candidates were tested on their knowledge of the types of problem that where different types of networks are applicable. Few candidates attempted this question.

Part (a) explored how the type of input data can affect the type of unit that is appropriate for a given application. For example, some unit types have outputs that are discrete whilst others have continuous outputs. Some problems have readily available training sets for supervised learning whilst others require the network to explore and come up with their own, as yet unknown, classifications.

Having decided on the types of network to use and trained their networks, application developers need to evaluate their results. This is often done poorly by those new to using neural networks, and the course gives advice in the form of some naïve predictors of time series against which to measure any proposed neural network that is claimed to predict such series. In part (b) candidates were tested on their knowledge of these. Given the importance of this issue it was disappointing to see the poor quality of the answers given.

Part (c) allowed candidates to display their knowledge of neural network applications, for example as studied as part of coursework. Examples of applications of both multi-layered feed forward networks and Kohonen-Grossberg counterpropagation networks were required.

Many answers lacked the details expected of good answers and so marks were not as high as they should have been. On the whole candidates seemed to know more about the applications of feed forward networks than Kohonen-Grossberg networks.

Few candidates attempted this question and overall most answers were poor.