Biologically Inspired Computation (F20BC/F21BC) Coursework

This assessment is done in two stages, each worth 25% of your overall course mark:

Stage 1 deadline: 3:30pm, Tuesday 26th October 2021 (Week 7)

Stage 2 deadline: 3:30pm, Tuesday 23rd November 2021 (Week 11)

Overview

This assessment aims to increase your understanding of both artificial neural networks (ANNs) and particle swarm optimisation (PSO), two biologically-inspired techniques which are taught in the course. Stage 1 involves implementing an ANN and then training it using backpropagation. Stage 2 involves implementing PSO and then using it to train an ANN.

Please read through the following important points before you begin:

- In order to encourage discussion of the topic, **the assessment involves working in pairs**, i.e., 2 people. You need to choose a partner from the same level (i.e., an F20BC student cannot be paired with an F21BC student). You should report your team choice on Canvas by **Wednesday 22nd September** using one of the following forms:
    - For Edinburgh students, go to Edinburgh Information → Edinburgh Team Formation
    - For Dubai students, go to Dubai Information → Dubai Team Formation
- Each member of a pair should contribute equally. We will be asking you to summarise your contributions when you submit each stage of the work, and we will rebalance the marks within a pair in cases where one member contributes substantially more than the other.
- **Do not copy code from the internet or from other students**. This is very easy for us to detect, and every year we have to report students to the academic misconduct board for plagiarism. This involves a lot of work for us, and it very stressful for students to go through this process, so please don't submit code that isn't your own without acknowledging its source!

This is assessed coursework. You are allowed to discuss this assignment with students outside of your pair, but you should not copy their work, and you should not share your own work with other students. We will be carrying out automated plagiarism checks on both code and text submissions.

Special note for re-using existing code. If you are re-using code that you have not yourself written, then this must clearly be indicated, making clear which parts were not written by you and clearly stating where it was taken from. If your code is found elsewhere by the person marking your work, and you have not mentioned this, you may find yourself having to go before a disciplinary committee and possibly face grave consequences.

Late submission and extensions. Late submissions will be marked according to the university's late submissions policy, i.e., a 30% deduction if submitted within 5 working days of the deadline, and a mark of 0% after that. The deadline for this work is not negotiable. If you are unable to complete the assignment by the deadline due to circumstances beyond your control (e.g. illness or family bereavement), you should complete and submit a mitigating circumstances application: https://www.hw.ac.uk/students/studies/examinations/mitigating-circumstances.htm

Stage 1

What you are asked to do:

1. Implement a multi-layer ANN architecture
2. Train the ANN to fit a specified dataset
3. Investigate how hyperparameters affect the ability of an ANN to fit this dataset
4. Write a short report and submit both the report and your code to Canvas
5. Sign the "Coursework Group Signing Sheet" and submit with your work

These tasks are described in more detail below. Implementation should be done using a language of your choice (e.g., Java, Python, Matlab, C, C++). The aim is for you to learn how to implement biologically-inspired approaches from scratch, so you should **not** use existing ANN libraries.

1. Implement a multi-layer ANN architecture

You should implement a simple feedforward multilayer architecture. Both the number of neurons in each layer and the number of layers should be configurable. You will need to implement both forward pass and backpropagation for the ANN. Here is a list of activation functions that should be implemented, though you may also investigate other suitable functions:

- Logistic function: $\frac{1}{1+e^{-x}}$
- ReLU (rectified linear unit): $\max(0, x)$
- Hyperbolic tangent: $\tanh x$

2. Train the ANN to fit a specified dataset

The problem domain for this work is binary classification. Your task is to build an ANN that fits the UCI banknote authentication dataset. You can download the dataset from the following link:
https://archive.ics.uci.edu/ml/machine-learning-databases/00267/data_banknote_authentication.txt

The dataset is provided in CSV format. It comprises 1372 records, each with 4 features (i.e. inputs to the ANN) and a classification (i.e. the desired output class, represented as either a 0 or 1 in this case). You can find more information here:
https://archive.ics.uci.edu/ml/datasets/banknote+authentication#

***Note:** Since not all students are taking the data mining and machine learning course, we do not require you to consider overfitting and thus you don't need to split the data into training and testing sets. You can just fit the ANN with the entire dataset and record the classification accuracy. Of course, if you have come across these concepts, you are welcome to go the extra mile by splitting the data and investigating overfitting of ANNs.*

Here are some points to consider:

- To evaluate how well an ANN fits the dataset, you should use classification accuracy; that is, the number of correctly classified data samples divided by the total number of data samples.
- The hyperparameters which determine the structure of an ANN include:
  - The number of input nodes in the input layer (of course, the problem determines this, you it should be configurable in your code)

    - The number of hidden layers
    - The number of hidden neurons in each hidden layer
    - The type of activation function used in each layer
    - The learning rate
    - The number of epochs for training
    - The type of loss function (at least cross entropy loss function, but feel free to use others). TIP: you can do a literature review and check which types of loss functions are suitable for classification.
- Your program should be able to create and train ANNs based on the above hyperparameters, as specified by a user or another program.
- Your program should record data that allows you to later visualise the graphs of the classification accuracy and training time against the hyperparameters, such as the number of epochs, learning rate, number of hidden nodes.
- To make the implementation of backpropagation easier, you can start from one hidden layer, and one output node (this is the basic requirement, see **Section 6: Marking Scheme**). You will get higher marks if you properly implement more than one hidden layer, more than one output nodes (say, two output nodes for binary classification), and more types of loss function.
- You are free to go further by adding more functionalities, and these efforts will be rewarded if they are properly done.

3. Investigate how hyperparameters affect the ability of an ANN to fit this dataset

ANN has a number of hyperparameters, as mentioned in Section 2, and these all influence the learning performance, including the learning accuracy and training time. For this part of the coursework, you should carry out an experimental investigation of how these hyperparameters affect the ability of ANNs for the binary classification task.

Here are some points to bear in mind:

- You can begin by informally trying out different values for the various hyperparameters and getting a feel for how much they affect the results.
- Pick some hyperparameters which you think have significant effect and then carry out a more formal experimental investigation.
- Pick a sensible range of values for each hyperparameter that you investigate. This could be guided by values you find in books, published papers etc.
- You can then perform systematic search for hyperparameters, for example, a random search on the hyperparameter space.
- There are random elements in ANN, for example, the initialisation of weight values, meaning that for the same problem and hyperparameter values, you may get different results each time you run it. Therefore, the distribution of results across a series of runs is more informative than the result of a single run. For example, it is common to give the mean result across at least 10 repeated runs and the standard deviation.

4. Write a short report and submit both the report and your code to Canvas

Your report should:

- Be no more than **4 pages** in length (max of 1500 words, not including references). You should take this into account when planning your experiments. If you have more results than you have space for, then pick the results that you think are most insightful and briefly mention which other experiments you carried out.

- Be written in Arial, or a similar font, with a minimum **font size of 12**.
- Briefly describe your implementations of ANN, noting any interesting aspects.
- Report the results of your experimental investigation. For instance, you might use tables that show the average results, and plots that show how hyperparameter values effect these averages.
- Referring to these results, discuss how the various hyperparameters affected the performance of your implementations of ANNs, and say why you think this is the case.
- Include useful references to the wider literature. For instance, you might use references to books or papers to justify particular implementation choices, or you could compare your findings to those reported elsewhere. Use standard referencing styles for this.
- Your report should contain the following sections: Introduction, Program development rationale, Methods, Results, Discussion and Conclusions, References.

You should submit both your report (as a **pdf** file) and your code (as a **zip** file) to Canvas using the links provided. Grading will use the assessment criteria given in the tables in Section 6.

5. Sign the "Coursework Group Signing Sheet" and submit with your work

The Coursework Group Signing Sheet is available on Canvas. Both members of your team should outline their contribution to the coursework, and sign the sheet. You may either include this sheet with your report, or include it in the zip file. **Note that no marks will be issued until we have this signed sheet.**

6. Marking Schemes

Note that the marking schemes are slightly different for F20BC and F21BC students, so make sure you look at the right one.

Marking scheme for F20BC Students: This assessment is worth 25**%** of the overall course mark**.**

Criteria	Weight	A (70-100%)	B (60-69%)	C (50-59%)	D (40-49%)	E/F (<40%)
Implementation ***(i.e. ANN code, comments and documentation)***	50%	Creative implementations of ANN that exceed the basic requirements. Correct evaluation code. Easy to read and well structured.	Correct implementations of the basic requirements. Generally good coding, structure and documentation.	Some significant issues in terms of correctness, structure, coding practice and documentation.	Major issues in terms of correctness, structure, coding practice and documentation.	Critical errors: for example, the code does not compile and/or run, or inappropriate algorithms have been implemented.
Experimental study ***(i.e. choice and validity of experiments performed, presentation of results)***	20%	Hyperparameters investigated are well motivated and their values are well chosen. Suitable results have been collected and are clearly presented and meaningful.	Some minor issues in terms of the motivation or description of hyperparameters, the experiments performed, or the presentation of results.	Some significant issues in terms of the motivation or description of hyperparameters, the experiments performed, or the presentation of results.	Some major issues: experiments do not make sense, have invalid results, or the study is not adequately described.	Some critical issues: experimental study is nonsensical or missing, inappropriate experiments, or the description of the study is uninformative.
Wider discussion ***(i.e. intro, interpretation***	20%	Clear, insightful discussion that shows a good understanding of	Generally clear and insightful, but shows some misunderstanding	The discussion is limited in terms of the depth or volume of	Some major issues in terms of depth or volume of	No real demonstration that the subject matter has

of results, conclusions, use of the wider literature or internet resources)		ANNs and includes well chosen references to the wider literature or internet resources.	of ANNs. Adequate use of the wider literature or internet resources.	understanding it demonstrates. Little or no use of the wider literature or internet resources.	understanding. No use of the wider literature or internet resources.	been understood, or very limited in its scope.
Report ***(i.e. structure, language, referencing etc.)***	10%	Report is well structured and divided into sections; good use of language; respects page limit and formatting guidelines	Report is suitably structured and divided into sections; mostly good use of language; respects page limit and formatting guidelines	Report is structured but not divided into sections; language issues that affect readability; notable formatting issues	Report is poorly structured; substantial language issues that affect readability; significant formatting issues	Report has a nonsensical structure; language issues make it very hard to read; problematic formatting

Marking scheme for F21BC Students: This assessment is worth 25**%** of the overall course mark**.**

Criteria	Weight	A (70-100%)	B (60-69%)	C (50-59%)	D (40-49%)	E/F (<40%)
Implementation ***(i.e. ANN code, comments and documentation)***	45%	Creative implementations of ANN that exceed the basic requirements. Correct evaluation code. Easy to read and well structured.	Correct implementations of the basic requirements. Generally good coding, structure and documentation.	Some significant issues in terms of correctness, structure, coding practice and documentation.	Major issues in terms of correctness, structure, coding practice and documentation.	Critical errors: for example, the code does not compile and/or run, or inappropriate algorithms have been implemented.
Experimental study ***(i.e. choice and validity of experiments performed, presentation of results)***	20%	Hyperparameters investigated are well motivated and their values are well chosen. Suitable results have been collected and are clearly presented and meaningful.	Some minor issues in terms of the motivation or description of hyperparameters, the experiments performed, or the presentation of results.	Some significant issues in terms of the motivation or description of hyperparameters, the experiments performed, or the presentation of results.	Some major issues: experiments do not make sense, have invalid results, or the study is not adequately described.	Some critical issues: experimental study is nonsensical or missing, inappropriate experiments, or the description of the study is uninformative.
Wider discussion ***(i.e. intro, interpretation of results, conclusions, use of the wider literature or internet resources)***	25%	Clear, insightful discussion that shows a good understanding of ANNs and includes well chosen references to the wider literature or internet resources.	Generally clear and insightful, but shows some misunderstanding of ANNs. Adequate use of the wider literature or internet resources.	The discussion is limited in terms of the depth or volume of understanding it demonstrates. Little or no use of the wider literature or internet resources.	Some major issues in terms of depth or volume of understanding. No use of the wider literature or internet resources.	No real demonstration that the subject matter has been understood, or very limited in its scope.
Report ***(i.e. structure, language, referencing etc.)***	10%	Report is well structured and divided into sections; good use of language; respects page limit and formatting guidelines	Report is suitably structured and divided into sections; mostly good use of language; respects page limit and formatting guidelines	Report is structured but not divided into sections; language issues that affect readability; notable formatting issues	Report is poorly structured; substantial language issues that affect readability; significant formatting issues	Report has a nonsensical structure; language issues make it very hard to read; problematic formatting

Stage 2

What you are asked to do:

1. Implement PSO
2. Use PSO to train an ANN
3. Do an experimental comparison between backpropagation and PSO
4. Write a short report and submit both the report and your code to Canvas
5. Sign the "Coursework Group Signing Sheet" and submit with your work

These tasks are described in more detail below. Implementation should be done using a language of your choice (e.g., Java, Python, Matlab, C, C++). The aim is for you to learn how to implement biologically-inspired approaches from scratch, so you should **not** use existing PSO or ANN libraries.

1. Implement PSO

You should implement a version of Particle Swarm Optimisation (PSO) that uses informants, i.e. each particle should be influenced by a subset of the other particles in the swarm, rather than just the swarm best.

Here are some points to bear in mind:

- How you allocate informants is up to you; for instance, they could be randomly allocated at initialisation. You might want to compare different strategies for allocating informants.
- You might find the pseudocode in the book "Essentials of Metaheuristics" useful as a starting point: https://cs.gmu.edu/~sean/book/metaheuristics/
- There are lots of variants of PSO. Whilst you are not expected to be aware of all of these, you are encouraged to read about PSO and experiment with different approaches. You will get more marks if you investigate multiple approaches or go beyond a basic version of PSO.

2. Use PSO to train an ANN

You should use the PSO code you wrote to optimise the weights of an ANN. You should solve the same problem as in Stage 1 (i.e., fitting an ANN to the specified dataset) but this time you are training the weights using PSO rather than backpropagation. This will involve coupling your PSO code to the ANN code you wrote in Stage 1.

Here are some points to bear in mind:

- Each particle within the PSO swarm represents an ANN as a fixed-length vector of floating-point values, each of which encodes the value of a particular weight within the ANN.
- Each time a particle is evaluated in PSO, the values in its vector should be used to set the weights of the ANN, and the ANN should then be evaluated against the dataset in order to measure its accuracy. This accuracy value then becomes the particle's fitness.
- Although there are versions of PSO that can handle variable-length vectors, you are not expected to know about these. Consequently, the architecture of the ANN (i.e., the number of layers and neurons) should be specified at the beginning of a PSO run and remain fixed.
- One advantage of using PSO, rather than backpropagation, is that you can optimise other aspects of the ANN in addition to the weights. For instance, you might also try encoding the activation functions used by the ANN within the PSO solution vector.

3. Do an experimental comparison between backpropagation and PSO

Now that you've implemented code for training an ANN's weights using both backpropagation (in Stage 1) and PSO (in Stage 2), you can compare which approach works better for fitting an ANN to this particular dataset.

Here are some points to bear in mind:

- In order to ensure a fair comparison, you should spend time optimising the hyperparameters of PSO, and you should report your findings. You can use a similar approach to hyperparameter optimisation to the one you used in Stage 1, but this time optimising PSO-specific hyperparameters such as the swarm size, the number of informants per particle, the acceleration coefficients etc. (see the PSO lecture in Week 7 for more on PSO hyperparameters).
- There's no need to go through the process of optimising ANN hyperparameters again, e.g., the number of neurons and hidden layers, since you should (from Stage 1) already have a good idea of which values work well. Also, to ensure a fair comparison between backpropagation and PSO, it makes sense to use the same ANN architecture for both.
- In addition to accuracy, you might also try to compare the two optimisation approaches in terms of how long they take to train an ANN.

4. Write a short report and submit both the report and your code to Canvas

This should follow a similar format to the Stage 1 report, but this time focusing on your implementation of PSO, and your experimental comparison between PSO and backpropagation. The same page limits, formatting guidelines and submission instructions apply.

5. Sign the "Coursework Group Signing Sheet" and submit with your work

You should submit a new Coursework Group Signing Sheet for Stage 2.

6. Marking Schemes

Note that the marking schemes are slightly different for F20BC and F21BC students, so make sure you look at the right one.

Marking scheme for F20BC Students: This assessment is worth 25**%** of the overall course mark**.**

Criteria	Weight	A (70-100%)	B (60-69%)	C (50-59%)	D (40-49%)	E/F (<40%)
Implementation ***(i.e. ANN code, comments and documentation)***	50%	A correct implementation of PSO that goes significantly beyond the basic specification. Correct evaluation code. Easy to read and well structured.	A correct implementation of PSO that meets or goes a little beyond the basic specification. Good coding, structure and documentation.	A basic implementation of PSO, or with significant issues in terms of correctness, structure, coding practice and documentation.	Major issues in terms of correctness, structure, coding practice and documentation.	Critical errors: for example, the code does not compile and/or run, or inappropriate algorithms have been implemented.
Experimental study ***(i.e. choice and validity of experiments performed, presentation of results)***	20%	A fair insightful comparison between PSO and backprop, important PSO hyperparameters have been optimised, results are clearly presented and meaningful.	A fair comparison between PSO and backprop, there has been some hyperparameter optimisation, results are mostly clear and meaningful.	The comparison has issues in terms of fairness, there has been little PSO hyperparameter optimisation, or there are some significant issues with the results.	Some major issues in terms of fairness, no hyperparameter optimisation, nonsen, invalid results, or the study is not adequately described.	Some critical issues: experimental study is nonsensical or missing, inappropriate experiments, or the description of the study is uninformative.

Wider discussion ***(i.e. intro, interpretation of results, conclusions, use of the wider literature or internet resources)***	20%	Clear, insightful discussion that shows a good understanding of PSO and its variants, and includes well chosen references to the wider literature.	Generally clear and insightful, but not very ambitious in scope or shows some misunderstanding of PSO. Adequate use of the wider literature.	The discussion is limited in terms of the depth or volume of understanding it demonstrates. Little or no use of the wider literature.	Some major issues in terms of depth or volume of understanding. No use of the wider literature.	No real demonstration that the subject matter has been understood, or very limited in its scope.
Report ***(i.e. structure, language, referencing etc.)***	10%	Report is well structured and divided into sections; good use of language; respects page limit and formatting guidelines	Report is suitably structured and divided into sections; mostly good use of language; respects page limit and formatting guidelines	Report is structured but not divided into sections; language issues that affect readability; notable formatting issues	Report is poorly structured; substantial language issues that affect readability; significant formatting issues	Report has a nonsensical structure; language issues make it very hard to read; problematic formatting

Marking scheme for F21BC Students: This assessment is worth 25**%** of the overall course mark**.**

Criteria	Weight	A (70-100%)	B (60-69%)	C (50-59%)	D (40-49%)	E/F (<40%)
Implementation ***(i.e. ANN code, comments and documentation)***	45%	A correct implementation of PSO that goes significantly beyond the basic specification. Correct evaluation code. Easy to read and well structured.	A correct implementation of PSO that meets or goes a little beyond the basic specification. Good coding, structure and documentation.	A basic implementation of PSO, or with significant issues in terms of correctness, structure, coding practice and documentation.	Major issues in terms of correctness, structure, coding practice and documentation.	Critical errors: for example, the code does not compile and/or run, or inappropriate algorithms have been implemented.
Experimental study ***(i.e. choice and validity of experiments performed, presentation of results)***	20%	A fair insightful comparison between PSO and backprop, important PSO hyperparameters have been optimised, results are clearly presented and meaningful.	A fair comparison between PSO and backprop, there has been some hyperparameter optimisation, results are mostly clear and meaningful.	The comparison has issues in terms of fairness, there has been little PSO hyperparameter optimisation, or there are some significant issues with the results.	Some major issues in terms of fairness, no hyperparameter optimisation, nonsen, invalid results, or the study is not adequately described.	Some critical issues: experimental study is nonsensical or missing, inappropriate experiments, or the description of the study is uninformative.
Wider discussion ***(i.e. intro, interpretation of results, conclusions, use of the wider literature or internet resources)***	25%	Clear, insightful discussion that shows a good understanding of PSO and its variants, and includes well chosen references to the wider literature.	Generally clear and insightful, but not very ambitious in scope or shows some misunderstanding of PSO. Adequate use of the wider literature.	The discussion is limited in terms of the depth or volume of understanding it demonstrates. Little or no use of the wider literature.	Some major issues in terms of depth or volume of understanding. No use of the wider literature.	No real demonstration that the subject matter has been understood, or very limited in its scope.
Report ***(i.e. structure, language, referencing etc.)***	10%	Report is well structured and divided into sections; good use of language; respects page limit and formatting guidelines	Report is suitably structured and divided into sections; mostly good use of language; respects page limit and formatting guidelines	Report is structured but not divided into sections; language issues that affect readability; notable formatting issues	Report is poorly structured; substantial language issues that affect readability; significant formatting issues	Report has a nonsensical structure; language issues make it very hard to read; problematic formatting