Intelligent Technologies COIY065H7
2016-2017
Coursework description, guidelines and marking scheme

This coursework is only for MSc students

1. Introduction

This assignment is an integral part of this module and contributes 20% to the overall mark. You should explore approaches that employ classifiers to solve a real-world classification problem. To this end, the module provides a set of patterns of textural descriptors for the detection of malignant regions in colonoscopy video frames. Colonoscopy is a screening procedure which includes direct visual examination of the colon by means of a fiberoptic endoscope. The coursework will investigate the performance of neural network architectures/training algorithms in this recognition task. Training and test patterns in ZIP format are available on Moodle.

If you have specific interest in a domain other than the provided dataset and you would prefer to use a dataset from that domain, e.g. physics, finance, marketing, bioinformatics etc., then you can choose a classification problem from the list maintained by the UC Irvine Machine Learning Repository- https://goo.gl/yj5nUU . Although I have used some of these datasets before in my own research and/or various student projects, I am not familiar with all of them- if you are planning to follow this route drop an email to gmagoulas@dcs.bbk.ac.uk to let me know which dataset you want to use and give me a couple of days to check it. Some of these datasets have not been used before, or they may be very large (in terms of features dimensionality or number of data points) so it may not be feasible to use them for this coursework- remember this is only a piece of coursework, not your MSc project!

Although you are allowed to use any programming language or software library for this assignment, the use of MATLAB Neural Networks Toolbox is suggested as it provides ready-made routines for building neural network architectures and training algorithms that are appropriate for the colonoscopy dataset provided.

If you are planning to use a software package or library, other than Matlab or R, make sure that you download the latest version and drop an email to gmagoulas@dcs.bbk.ac.uk to let me know.

The assignment is explained in Sections 2-4 below. Section 5 of this document gives you an example of how to structure your report and explains the marking scheme. Section 6 presents the deadlines and submission instructions. Section 7 explains the penalties for late submissions, and Section 8 explains how the College deals with plagiarism. Sections 9-11 provide additional information on learning resources, referencing, and exploitation of results.

2. Detection of lesions in colonoscopy: the problem and the data

Colonoscopy is the most accurate screening technique for detecting polyps, also allowing biopsy of lesions and resection of most of the polyps. The procedure is carried out by an expert who interprets the physical surface properties of the tissue- such as the roughness or the smoothness, the regularity, and the shape - to detect abnormalities. Adjacent surfaces of the colon lining showing different properties are distinguished on the basis of the textural variations of their tissue (the texture of the tissue is considered as a composition of pit patterns). These textural alterations of the colonic mucosal surface signify that this property could also be used for the automatic detection of lesions.

The approach followed to generate the dataset consists of two processing stages. The first stage consists of procedures that are applied to the frames of the video sequence to extract all the identifiable features, which form the feature vectors. To this end, a family of texture attributes that correspond to the components of the feature vectors and account for the main spatial relations between the grey levels of the texture has been chosen. The particular method applied is called concurrent matrices and produces 16-dimensional features vectors for each region (rectangular windows of 64 x 64 pixels) of the frame is applied on.

The second processing stage, which is the focus of this assignment, decides on the image regions characterisation. To this end, several researchers have proposed techniques that range from linear discriminant analysis to sophisticated AI detection schemes that are based on artificial neural networks, support vector machines, multiple-classifier, fuzzy or neuro-fuzzy systems.

For this assignment, you are provided with a set of data files that contain patterns (feature vectors) extracted from a short endoscopy video sequence of six frames. The dataset is organised into training and test data files; there is a training data file and a test data file for each of the six frames.

Files with training data are named as `framenumber`trn.ssv (e.g. 1trn.ssv, 4trn.ssv), whilst files with test data are named as `framenumber`tst.ssv (e.g. 1tstn.ssv, 4tst.ssv). These are ASCII files that contains train/test data and can be opened using a standard editor, e.g. Wordpad, MS-Word, or using MS-Excel.

In these files, each row represents a training or testing pattern and consists of 17 elements (columns). The first 16 elements correspond to features of a particular frame region without any normalisation. The last element is either 0 or 255, where a value of zero indicates that the corresponding vector represents a normal tissue sample while a value of 255 indicates that the pattern is associated with an abnormal tissue sample and in your programs it should be substituted by a value of 1.

This should normally lead you to use 16-dimensional vectors for input data and 2-dimensional vectors for the desired output (formulating the problem as a 2-class problem, i.e. normal/abnormal defined as 0/1). Typically a normalisation procedure is applied on the training and testing data to bring their values in the range of [0,1] or [-1,1] depending on the application.

For training, you should use vectors from the `framenumber`trn.ssv files and for testing vectors from the `framenumber`tst.ssv.

Further details on the problem and the use of classifiers in this context can be found in the paper:

Magoulas G.D., Plagianakos V.P., and Vrahatis M.N., Neural Network-based Colonoscopic Diagnosis Using On-line Learning and Differential Evolution, Applied Soft Computing, Vol. 4(4), 369-379, 2004. Available online at: http://www.dcs.bbk.ac.uk/~gmagoulas/ApplSoftComp.pdf

3. Experimental investigation on the performance of your classifier

Although you can attempt to use any classifier you wish, here we concentrate on neural networks that can be trained by minimising a measure related to network's performance, such as the learning error which can be defined by the sum of the squared difference between the actual output vector of the network and the desired one over the whole training set. This approach is very popular in neural network training and includes learning algorithms that operate off-line, also called batch learning, or on-line learning, also called pattern-based learning.

A classifier is normally trained to achieve up to 97% of success in discriminating between normal/abnormal patterns. For example, if the training set has 1200 (say 600 normal + 600 abnormal) patterns then a network has been trained successfully when is able to categorise at least 1164 out of the 1200 patterns. When real-world noisy data are used a lower success of 90% or even lower, e.g. 80%, in training might also produce good results in testing – this depends on the level of noise in the data and it is difficult to know in advance, i.e. before doing some attempts to train the networks.

The trained classifier is tested over the entire test set (use data from the six frames stored in the "?tst.ssv" files).

You should do the following experiments:

1) Explore the effect of the structure of the classifier on the performance. For example, if you use feedforward or recurrent networks vary the number of hidden nodes from 5 to 55 in steps of 10. For each number of hidden nodes, use the six training files to train 30 networks starting from the same random initial weights using one of the following learning algorithms: Scaled conjugate gradients (SCG), Levenberg- Marquardt and Rprop (all these learning algorithms are already available in Matlab). Store in ASCII format for each frame: the number of successfully recognised normal patterns, the number of unsuccessfully recognised normal patterns, the number of successfully recognised abnormal patterns, the number of unsuccessfully recognised abnormal patterns, the overall classification success in training, the number of iterations required (epochs) to reach the training goal (ideally 90% success in training) and the sum of the squared error in training (i.e. the sum of the squared difference between the actual output vector of the network and the desired one over the whole training set).
2) test each one of the above trained classifiers with the test data and store in ASCII format for each frame: the number of successfully recognised normal patterns, the number of unsuccessfully recognised normal patterns, the number of successfully recognised abnormal patterns, the number of unsuccessfully recognised abnormal patterns, the overall classification success in testing, and the sum of the squared error in testing (i.e. the sum of the squared difference between the actual output vector of the network and the desired one over the whole test set).

So you train and test 6x30 networks ("number of hidden nodes configurations" x "number of networks trained") for each frame. *Training should stop when your networks achieve a 97% success in discriminating between normal/abnormal patterns in the training set (or lower if this is not possible but explain in your report the reasons).*

The results of your experiments should be stored in ASCII format (see above) in separate files - one file for each frame - specifying whether the result is from the training or the testing phase, and should be submitted together with your report. Check that these ASCII files can be opened with a simple text editor, such as notepad or wordpad, or Excel.

4. Implementation issues

You can implement your classifiers in MATLAB or write your own code or use a package/library from the internet; I wouldn't recommend implementing everything from scratch unless you are very experienced with Java, C++ or some other programming language. In all cases, make sure that all sources and code taken by others or the internet are cited properly in your final report otherwise you may be accused for plagiarism. If you are planning to use an open source library or any kind of software other than Matlab or R, drop an email to gmagoulas@dcs.bbk.ac.uk to let me know.

Although Levenberg-Marquardt; SCG, and Rprop are mentioned in Section 3 as the algorithms for experimentation, some packages may provide other methods. Moreover, some packages provide techniques for determining the optimal structures of classifiers (e.g. constructive neural networks) automatically as part of the training. In that case, instead of performing

experimental tests varying the number of parameters, e.g. number of hidden units as mentioned in Section 3, these techniques can be used to find the appropriate structure for your classifiers.

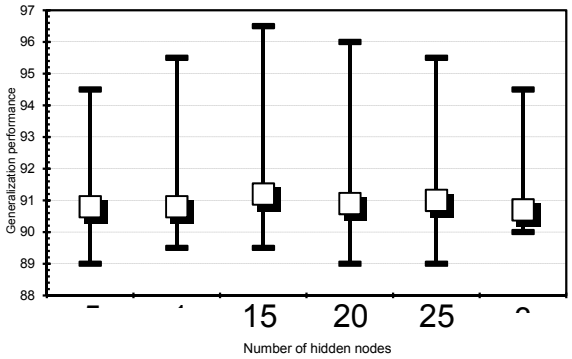
Note that your comparisons would be more meaningful, if a validation technique is used, such as k-fold cross validation (k=7 or k=10 is typically used), or leave-one-out cross validation if the training set is small. Lastly, the use of weight decay or weight elimination (also called regularisation is some software packages) could help you to get better results in successfully recognising normal/abnormal patterns.

5. Assignment outline and marking scheme

Your work will be presented in a report (about 3000 words). It is important that your report is properly structured. Sections like the ones shown below should be included in your report to ensure good coverage of the topic.

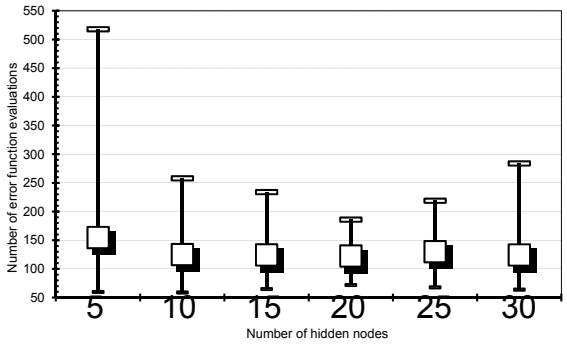
1. Methods used (30% of the mark)
    - 1.1 This part should normally describe clearly the classifier architectures used in your assignment and any relevant parameters (e.g. number of hidden nodes, activation functions of neural networks etc).
    - 1.2 This part should describe the algorithm used for training and their parameters. For example if you use Rprop backpropagation then initial learning rate values used should be stated. Also explain any normalisation techniques used and whether you have used some form of cross-validation or weight decay, providing details of the particular method.
2. Experiments, findings and discussion (60% of the mark)

    In this part you should provide a detailed account of your experiments and results (see Section 3), and discuss your findings. You can use Excel to provide charts - like the figure below, which uses error bars (Box and Whisker Charts in Excel), to show the performance of neural networks in terms of generalisation with respect to their number of hidden nodes – presenting and discussing your results.



    Alternatively, one could use tables to provide the same information by giving for each number of hidden nodes the average value, the minimum value, and the maximum value of generalisation performance (in percentage of successfully recognised patterns) in the tests.

    You can also discuss about the cost of the computations, e.g. referring to the number of training iterations required or the number of error function evaluations (see figure below)



For example, the average classification success for neural classifiers can be also discussed with respect to the number of hidden node used and training algorithm. Results can be also presented in tables like the one below that shows average performance in terms of recognition success for two methods in one of the frames of a video sequence.

FRAME 4

Method	Cancer (%)	Normal (%)	Mean (%)
Method 1	83	96	93
Method 2	73	93	88

3. Conclusions (5% of the mark)
   - 3.1 Provide an overview/summary of your work and findings.
   - 3.2 Identify areas for improvement; discuss what you could have done better (particularly important if you failed some of your targets).
4. Bibliography (5% of the mark)

   Provide a list of the bibliographical/web sources you used. Include publication details and all information necessary to access the online resources. Sources should be cited in the text by (Author name, year) and appear in the references list in alphabetical order by Author's last name. This also applies to websites, e.g. an online article/webpage should be listed in your references; for example:
   (MLOSS, 2011). *Machine learning open source repository. Available online at http://mloss.org/*

NOTE: use of any text or code (even open source code) taken from other sources should be clearly identified and referenced in your report to avoid plagiarism (see Section 8). If you are unsure on which parts of your code needs appropriate referencing do consult the module lecturer.

6. Deadlines and submission instructions

Submission is only done electronically through Moodle and consists of the submission of a report, data files with your results and code (if existing Matlab toolboxes have been used, these should be mentioned in the report). Make sure you are familiar with Moodle and able to upload your files (for example you could test the system by uploading a test file). **Hardcopy versions of the report/data/code files will not be accepted.** The code should be included in the electronic copy of the report that you will submit as a Word document or RTF. **This is required as the text and the code are tested for plagiarism.**

You should upload on Moodle the completed assignment in **MS-Word for Windows format** or **MS-Word for Windows compatible format** by **January 20, 2017 at 11:00pm** (this is Moodle time not your PC's time. In case you are planning to upload your files whilst at a remote location make sure you check Moodle's time and take into account time zone differences).

Your files should be named according to your last name.
The first page of your Word document **MUST** have the following information:

Module title and code: Intelligent Technologies- COIY065H7
Name: your first name and last name
Emails: please provide your College email and the email you use- if different from your College email

Your report should have an Appendix with a description of data files and code submitted.

It is your responsibility to ensure that files transferred from your own machines are in the correct format and that any programs execute as intended on Department's systems prior to the submission date. Any specific instructions on software use should also be included in an Appendix of the report, entitled "Instructions for using the code".

Each piece of submitted work MUST also have a page entitled "Academic Declaration" by the author that certifies that the author has read and understood the sections of plagiarism in the document http://www.bbk.ac.uk/mybirkbeck/services/rules/Assessment%20Offences.pdf that describes College's Policy on assessment offences. Confirm that the work is your own, with the work of others fully acknowledged. Submissions must also be accompanied by a declaration giving us permission to submit your report to the plagiarism testing database that the College is using.

Reports without a Declaration form are not considered as completed assignments and are not marked.

The Academic Declaration should read as follows: "I have read and understood the sections of plagiarism in the College Policy on assessment offences and confirm that the work is my own, with the work of others clearly acknowledged. I give my permission to submit my report to the plagiarism testing database that the College is using and test it using plagiarism detection software, search engines or meta-searching software."

You should note that all original material is retained by the Department for reference by internal and external examiners when moderating and standardising the overall marks after the end of the module. You will receive a grade and feedback through Moodle 21 working days after the cut-off deadline (see Section 7).

Those who would like to get some early feedback on their assignment before submitting the completed work, they can email <gmagoulas@dcs.bbk.ac.uk> a draft of their report for comments. This should be done **by December 18, 2016** as we will not be able to comment on drafts sent after this date.

7. Late coursework

It is our policy to accept and mark late submissions of coursework. You do not need to negotiate new deadlines and there is no need to obtain prior consent of the module lecturer.

We will accept and mark late items of coursework up to and including seven working days after the normal deadline. Therefore the **last day the system will accept a late submission for this module is January 29, 2017 at 11:00pm** (this is Moodle time not your PC's time. In case you are planning to upload your files whilst at a remote location make sure you check the Blackboard time and take into account time zone differences). **January 29, 2017 at 11:00pm is the absolute cut-off deadline for coursework submission.**

However, penalty applies on late submissions. Thus the maximum mark one can get in the coursework is 50%. If you believe you have good cause to be excused the penalty for late submission of your coursework, you must make a written request using a mitigating

circumstances application form and attach any evidence. Your form should be handed in or emailed to the MSc Programme Administrator (with a carbon copy to the module lecturer and the Programme Director) as soon as possible, ideally that is by the cut-off deadline. This letter/email does not need to be submitted at the same time as you submit the coursework itself but **MUST be submitted by February 11, 2017.**

Even if the personal circumstances that prevented you from submitting the coursework by the last day (i.e. February 11, 2017) are extreme, **the Department will not accept coursework after this date**. We will, naturally, be very sympathetic, and the MSc Programme Director will be happy to discuss ways in which you can proceed with your studies, but please do not ask us to accept coursework after this date; we will not be able to as there is a College-wide procedure for managing late submissions and extenuating circumstances in student assessment. As soon as you know that you will not be able to meet the deadline, it will be useful for you to discuss this with the module lecturer. They will be able to advise you on how best to proceed. Another person to speak to, particularly if the problem is serious, is the MSc Programme Director. You will then have the opportunity to discuss various options as to how best to continue your studies.

Further details concerning the rules and regulations with regard to all matters concerning assessment (which naturally includes coursework), you should consult College Regulations at http://www.bbk.ac.uk/mybirkbeck/services/rules. **Please see the 2016/17 programme booklet for the rules governing Late Submissions and consideration of Mitigating Circumstances and the Policy for Mitigating Circumstances at the College's website** http://www.bbk.ac.uk/mybirkbeck/services/rules.

8. Plagiarism

The College defines plagiarism as "copying a whole or substantial parts of a paper from a source text (e.g. a web site, journal article, book or encyclopedia), without proper acknowledgement; paraphrasing of another's piece of work closely, with minor changes but with the essential meaning, form and/or progression of ideas maintained; piecing together sections of the work of others into a new whole; procuring a paper from a company or essay bank (including Internet sites); submitting another student's work, with or without that student's knowledge; submitting a paper written by someone else (e.g. a peer or relative), and passing it off as one's own; representing a piece of joint or group work as one's own".

The College considers plagiarism a serious offence, and as such it warrants disciplinary action. This is particularly important in assessed pieces of work where the plagiarism goes so far as to dishonestly claim credit for ideas that have been taken from someone else.

Each piece of submitted work MUST have an "Academic Declaration" form signed by the student(s) which certifies that the students have read and understood the sections of plagiarism in the College Regulation and confirm that the work is their own, with the work of others fully acknowledged. Submissions must be also accompanied by a declaration giving us permission to submit coursework to a plagiarism testing database that the College is subscribed.

If you submit work without acknowledgement or reference of other students (or other people), then this is one of the most serious forms of plagiarism. When you wish to include material that is not the result of your own efforts alone, **you should make a reference to their contribution, just as if that were a published piece of work**. You should put a clear acknowledgement (either in the text itself, or as a footnote) identifying the students that you have worked with, and the contribution that they have made to your submission.

9. Referencing

References include the full bibliographic information about the source, such as the author(s)'s name(s), date of publication, title of work, place of publication, and publisher. This information is usually given in the section called Reference List or Bibliography at the end of the text. The key principle is that you should give enough information to allow another person to find the source for themselves.

Here are some examples using the Harvard referencing system:

[when you are referring to a book]
Lewin, K., 1951. Field Theory in Social Science. New York: Harper and Row.

[when you are referring to a chapter in a book, where 'ed.' means editor, and 'edn.' means 'edition']
Piaget, J., 1970. Piaget's theory. In: P. Smith, ed., Handbook of child psychology. 3rd edn. New York: Wiley, 1970, pp. 34-76.

[when you are referring to a journal article]
Holmqvist, M., 2003. A Dynamic Model of Intra- and Interorganizational Learning. Organization Studies, 24(1), 95-123.

[when you are referring to a webpage]
W3C, Web Accessibility Guidelines and Techniques, available online at http://www.w3.org/WAI/guid-tech.html. Last accessed 12/02/2015.

Independent of their type (e.g. book, article, webpage), all references are included at the end of a document in alphabetical order starting from the author's name as in the example above.

10. Useful resources

Here are some resources on plagiarism, study skills, time management and referencing that can help you to better manage your project and avoid plagiarism.

On Plagiarism

- https://owl.english.purdue.edu/owl/resource/589/1/

On Referencing Systems

- Harvard guide to citing references Available to online at: http://www.open.ac.uk/libraryservices/documents/Harvard_citation_hlp.pdf

On Study Skills

- http://www.brad.ac.uk/acad/management/external/els/informationsheets.php

11 Exploitation of coursework outputs

Students should consult the College's "Financial Regulations and Procedures" regarding IPR (http://www.bbk.ac.uk/fin/reporting/financial_regulations/finreg/index_html).

This document states that:

"Section E 14.2.1 (ii) Except as otherwise as agreed in writing, if a student in the course of studies, produces any original works (including computer software) which may be commercially exploitable, the College shall be entitled to the copyright in such works and shall

use its best endeavours to secure royalties. These will be shared as set out in the detailed code of practice".

These same document also state: "Students are required to comply with the College procedures for notifying any invention, device, material, product or process, computer software or other potentially valuable result which it is considered might have commercial significance, whether patentable or not, developed or invented during the course of students' research or study at the College and make assignment of rights to the College by signing a Deed of Assignment at the start of their project. Deeds of Assignment must be obtained from the College Secretary's Office, from where further copies of these College procedures, which include details of the sharing of revenue from exploitation, are available."