Title of PHY480 report

160207219
Department of Physics & Astronomy, University of Sheffield

4 May 2020

ABSTRACT

This LATEX document serves as a template for the PHY480 Semester II report in the format of MNRAS (Monthly Notices of the Royal Astronomical Society). The report should have a short abstract, up to about 200 words long, which summarizes the contents of the report, briefly describing its aims, methods, and main results. It should be a single paragraph, and no references should appear in the abstract.

1 INTRODUCTION

A key issue regarding modern day survey astronomy is the sheer amount of data available. The first step given any survey imagery is to determine which objects are of interest, and which are not. Previously this would be a very time consuming process of manual selection. With the advent of machine learning, a very real possibility of using a computer to perform this task arose.

Historically a need existed to classify astronomical object, from distinguishing between types of stars[cite here] to different types of nebulae[cite here] and later galaxies[cite here]. This process was usually very labor intensive and involved the time of many a grad student. That changed in the latter part of the 20th century. Starting with applying the AutoClass algorithm to the IRAS catalogue in 1988[cite here], astronomers started using computers to solve this classification problem using machines.

Early methods for this involved selecting features manually and separating the feature-selected variables into classes. This however, was still quite labor-intensive and did not solve the fundamental problem which was human bias. Human bias exists within these kinds of methods because the features selected are often based on the understanding of the researchers themselves. Therefore any feature selected may be the tainted by the human selection of variables. An example of this was Abraham and Merrifield 2000 [cite this], where they extracted features of galaxies which were based on their understanding—the features were early/late-type galaxies based on the Hubble sequence, and the strength of the central bar. While this report does not seek to discredit that paper, one should have a certain expectation that the results from this would then correspond to the Hubble 'tuning-fork' diagram due to the parameters selected—which it does.

Next in the development of this field was the advent of PCA (principle component analysis) and SVM (support vector machine)—a lengthier discussion may be found here [cite myself]. PCA enables feature selection that is blind to the researchers. This is because PCA essentially selects the principle components, which are linear combinations of the input features, and orders these principle components in order of variance (which can also be understood as information density). With this, one only needs to select as many principle components as one has the computing power to handle. This gives an objective way of both selecting features and dimension reduction. Using these principle components, it is possible to apply SVM to this now dimension-reduced dataset to classify the data if it is linearly separable. [cite papers which use this] However, there are still some shortcomings with this method. Most importantly, SVM is only effective on linearly separable data, and real life data often is not. Therefore, more sophisticated methods such as the 'kernal-trick' [cite this] may be used to try and separate the data. However even with these methods to extend the applicability of SVM, it remains ineffective on some datasets. There are other classification algorithms such as logistic regression, but they continue to lack to classify all datasets generality [cite?]. Another downfall is with PCA. Due to the nature of dimension reduction, some data must necessarily be discarded. Even though the PCA algorithm guarantees that the minimum amount of data will be discarded, some is discarded nonetheless. If one's computing power allows, it is clearly superior to not discard any data at all.

The next step in this field happened relatively recently—within the last 5 years—and is mainly due to advances in hardware as well as breakthroughs in computer science. The use of a neural network has been discussed for some time before this [cite here], in fact a SVM classifier can be seen as a prototype of neural networks. However, for a long time, deeper neural networks were hampered by a lack of ability to train them effectively due to convergence issues with optimisers.[cite here] When this issue was solved, the field of machine learning was also aided by the exponential growth of processor speed (viz. Moore's Law), which allowed more complex (deep) neural networks to be designed and trained. However, this field is still lacking the push to prominence that the field of computer vision has today.

That final push came in two forms: the unstable-gradient problem (also known as the the vanishing-gradient problem in certain instances, further discussion in Theory section below) was solved using further backpropagation (explained below) because hardware advances allowed this; and a shift of activation function because sigmoid functions sometimes cause a saturation in the final hidden layer [2010 Glorot and Bengio], this was remedied by using another activation function (ReLu function) that is ubiquitous in modern neural networks. This allowed deep convolutional neural networks to be used efficiently and thereby resulting in the rapid development of the field of Computer Vision. [Neural Networks and Deep Learning Book cite]

The proximate cause for this project and hence report is that the Gravitational wave Optical Transient Observatory (GOTO) records a very large amount of data which needs to be classified. Because

the amount of data is too large for any human to complete in any reasonable amount of time, machine learning was enlisted to solve this problem.

The primary goal of this project is to create a classifier that could classify sources on the image into stars or galaxies. The secondary goal is to extend that classifier to determine between spiral and elliptical galaxies.

This project approached this problem from a largely computer science perspective and therefore the techniques used stem from current techniques in computer science.

2   THEORY

Given that the project is a synthesis of the fields of survey astronomy and machine learning. Many of the terms and techniques used will be unfamiliar to astronomers who have not worked in the machine learning field. This section seeks to give an overview of the important parts which are necessary to understand this project.

2.1   Classifiers

A classifier is defined as an algorithm which predicts the class of a datapoint. A classifier can be seen as approximating a mapping function f from input variables X to discrete output variable y. N.B. In machine learning, these variables may be matrix or tensor objects. A datapoint here simply means an instance of X which the classifier needs to classify. [Towards data science https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623] Within this project, the datapoint is a single image cutout, and the output is a determination whether the image is a star or a galaxy.

There are many classifiers in existence, broadly speaking there are placed into two categories: Lazy-learners, which only classify objects when a test object is required to be classified; and Eager-learners, which trains a model based on training-set data and then applies the model onto testing data. Artificial neural networks—which are used in this project—are eager-learners. Eager-learners have the advantage being very quick to apply to new data at the expense of a lengthy training process.

Since simple classifiers are both classifiers in their own right, as well as acting as activation functions for more complex neural networks.

2.1.1   *Logistic Regression/Sigmoid Activation Function*

The logistic regression algorithm attempts to split a dataset into two classes—a positive class represented by an output of 1, and a negative class represented by an output of 0. It uses the sigmoid function to do so.

$$h(X) = \sigma(Z), \text{ where } \sigma(Z) = \frac{1}{1 + e^{-Z}} \tag{1}$$

h is a linear parameter which broadly speaking gives the probability that 'X' is in the positive class. From the above, it is clear that $\lim_{Z \to +\infty} \sigma(Z) = 1$ and $\lim_{Z \to -\infty} \sigma(Z) = 0$. This allows for the classification of X based on inputs to Z.

Given that, as stated above, X is a generalised variable representing all the features(variables) $\{x_1, x_2, x_3, ...\}$ that are inputs to the sigmoid function. Therefore X is a (usually column) vector.

The formal definition of the sigmoid function in the context of logistic regression is

$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}} \tag{2}$$

In Machine Learning the variable θ often denotes the weights applied to the generalised variable X to correctly predict the outcome. The θ^T in the exponent denotes the transpose of θ because θ is a column vector like X and so to multiply each weight by the appropriate variable, the transpose must be taken. N.B. An equivalent operation here is the dot product $\theta \cdot X$, however this is not usually used because it generalises poorly when multiple output y are needed, i.e. when

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix}.$$

The precise details on how to train a logistic regression model is beyond the scope of this report, but broadly speaking, a θ is found such that it correctly predicts the class of any given X.

Given that the sigmoid function is just a function that takes inputs from $-\infty$ to $+\infty$, and outputs a number between 0 and 1. In the case of logistic regression, this can be seen as a probability that the input belongs to either class, in more complex neural networks, this function can play the role of an activation function. An activation function is just a function which processes the inputs taken from one or multiple nodes and outputs it to some other node. The activation function itself resides within a node and has some node weight θ related to it.

A sigmoid function was ubiquitous as an activation function before 2011 because it satisfies a few requirements that make such an activation function very useful. It is nonlinear, which enables a two layer network to be a universal approximator [cite this], it is continuously differentiable which makes a gradient descent approach—a very common way of optimising a neural network—possible. However, it fails in some circumstances such as saturating the final hidden layer [2010 Glorot and Bengio]. Therefore the field of machine learning has steadily shifted away from using sigmoid activation functions.

2.1.2   *Rectifier/Rectified Linear Unit*

A rectifier is just like a sigmoid function in that it serves as an activation function in a node. The rectifier function is defined

$$f(x) = \max(0, x) = \begin{cases} 0, & \text{for } x \leq 0 \\ x, & \text{for } x > 0 \end{cases} \tag{3}$$

This function was found in [2010 Glorot and Bengio] and it quickly became the favoured choice of activation function for neural networks. In the same paper, it was shown that this function also satisfies the non-linearity requirement. This function is not differentiable everywhere though, its derivative at 0 is undefined. However there are two solutions to this: the analytic solution is to define the derivative as 0 or 1, since those are the values of the derivative of the function elsewhere—0 for $x < 0$, 1 for $x > 0$. If this derivative at 0 is defined, then the derivative for the function is continuous. The second solution is that in a practical context, a node would never reach 0, because of the nature of gradient descent, the local minima is never actually reached, there is always some small value above or below 0, therefore the issue of an undefined derivative is not reached. Finally, this activation function has the added benefit of having an infinite range: $(0, \infty]$, which increase the speed of convergence of a model.

2.2 Neural Networks

A neural network is a set of nodes (which include the aforementioned activation functions) arranged in a certain way. Pertinent to this project is the feedforward neural network (FNN) and the convolutional neural network (CNN) as those are the networks used.

2.2.1 Feedforward Neural Networks

A feedforward neural network (FNN) is a neural network only feeds data forward through the network, i.e. there are no loops. This is the most simple type of neural network but it is also the most versatile. Because the inputs can be any type of numerical data, of any dimension, the neural network can approximate anything (Universal Approximation Theorem) [cite]. Formerly, an FNN may only have 1 hidden layer which is due to both problems with training a neural network with anything more than 1 HL (backpropagation [cite]), but also due to nomenclature: anything with more than 1 HL is called a deep feedforward neural network. With the advent of much more powerful hardware and also techniques to effectively train multi-layered neural networks, these 'deep' feedforward neural networks replaced single-layered neural networks in almost all applications. Thence, all of these neural networks were generically termed 'Feedfoward Neural Networks'. [Not sure which parts can be cited, but probably find some recent online textbook somewhere]

[Insert diagram of generic FNN]

2.2.2 Convolutional Neural Networks

A convolutional neural network (CNN), strictly speaking, can be thought of as a type of FNN due to it not having any recurring loops. However, in practice it is generally thought of as distinct due to it having vastly different node behaviour.

A CNN, like its name suggests, is a neural network which has nodes which perform convolutional operations. Convolution is most often approached in physics in terms of a signal being convolved with a background. This is essentially the same operation that a convolutional node performs. A convolutional filter is applied to an image, this consists of a kernel which describes the function—which is almost always a linear shape—that passes over the original image and passing the result into a new image.

[Find picture of convolution]

A CNN has 3 types of layers: convolution layers, pooling layers, and a fully connected layer. A convolution layer performs a convolutional operation as described above. It has 4 key parameters: kernel, stride length, layer size, and activation function. The kernel describes the kernel which is passed over the incoming image, the key parameter of the kernel is the size—kernels are usually squares of 3x3, 5x5, or 7x7 size, with the size determining how many neighbouring pixels the filter wants to take into account. The stride length is how many pixels the filter jumps over before the next filter. The layer size is how many convolutional filters are passed over the incoming image. Because each nodes has different initial conditions, this represents the trainable part of the network. Finally, the activation function determines what value is outputted to any particular pixel. In a modern CNN it is usually ReLU due to it working better with sparse models. [cite ReLU sparse model, CNN]

A pooling layer can be thought of as a specific type of convolutional layer in that the kernel is a flat one. Its main purpose is to reduce the dimension of the incoming image.

[Insert pooling layer image]

There are two types of pooling layers: average pooling and max pooling. An average pooling layer inputs the average value of the pixels within the pooling filter and outputs them to the next image. A max pooling layer does essentially the same but with the maximum value of the pixels rather than the average value. This achieves the same amount of dimensionality reduction except the max pooling layer achieves noise suppression by discarding the noisy activations. The max pooling layer is therefore said to be superior to the average pooling layer. [Maybe cite tds https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53].

A fully connected layer is simply a layer from an FNN used to gather the results of the convolutional and pooling layers and output to a visible layer, of which the results can be used. In binary classification cases a binary activation function such as sigmoid, ReLU, or tanh can be used, while in multiple-class classification problem a function like softmax should be used instead. [cite softmax]

2.3 Optimisation algorithms

In order for a neural network to be of any use to anyone it must be trained. Training is a process where the weights of the neural network are modified to better predict the desired outcome. The broad idea behind optimisation of a neural network is that one is trying to minimise some loss function. Given that the problem is a minimisation problem, it would make sense to use some kind of algorithm which takes into account the gradient of the hypersurface that the optimiser is attempting to optimise over because it is known that at local minima the gradient of the surface is 0.

In a general form, the equation for gradient descent is

$$\theta_{n+1} = \theta_n - \alpha \nabla J(\theta) \tag{4}$$

where θ is a general set of weights applied at any particular node. θ_n is the set of weights before updating while θ_{n+1} is that after updating. α is a variable called the learning rate, which controls how 'far' in any one direction one update moves the weights on the hypersurface. ∇J is the key term here as this is the gradient of the cost(loss) function over all variables which ensures all the weights for the variables are updated in the same pass. [probably cite the Stanford course]

2.3.1 Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is an algorithm which chooses a random datapoint within the whole dataset to perform the gradient descent algorithm. This was developed due to the need to train larger and larger datasets. Larger datasets present a challenge due to the longer time it takes to calculate losses through the entire dataset, at some point large datasets present a hardware constraint of not having enough RAM to store the entire dataset in memory which makes gradient descent through the whole dataset unworkable.

SGD solves this by updating the weights after a single datapoint. This allows the weights to converge much quicker than in normal gradient descent. However, because each datapoint updates the gradient this introduces a lot of noise which cannot be removed. A modern compromise is to use mini-batches—updating the weights after every few 10s or 100s of datapoints. [SGD paper somewhere]

2.3.2 Momentum

A modern advance to the gradient descent algorithm uses a concept called momentum. The full mathematical treatment of momentum

is beyond the scope of this report, but a general explanation of the concept will be given. [cite or put it in appendix?] The general concept of momentum is the addition of a variable v, representing the momentum of the descending 'particle'. The updating equations are changed to

$$\begin{cases} v_{n+1} = \eta v_n - \alpha \nabla J(\theta) \\ \theta_{n+1} = \theta_n + v_{n+1} \end{cases} \quad (5)$$

with v being the new momentum parameter and η being a new decaying hyper-parameter. This has the effect of both accelerating convergence due to the momentum term accumulating a large value from repeated updates in the same direction; and the effect of smoothing out noise because local fluctuations in gradient get smoothed over by the overriding momentum term.

A variant of the momentum approach is the Nesterov Accelerated Gradient (NAG). The principle is essentially the same except the algorithm calculates the momentum using past values i.e. v_n and v_{n-1} in lieu of v_{n+1} and v_n respectively as well a slightly modifying the gradient calculation. This method has a key benefit of being able to 'slow down' the momentum before it reaches the minimum of the hyperplane thereby reducing the possibility of 'overshooting'.

2.3.3 Modern Optimisers

Modern optimisers tend to use an adaptive learning rate.

[liberally get citations from here https://ruder.io/optimizing-gradient-descent/index.html]

The learning rate α as described above in the aforementioned optimisers are constants. However it is often advantageous for the learning rate to change with respect to the parameters.

The first modern optimiser in this regard is Adagrad (for 'adaptive gradient') published in 2011. [cite paper] This implementation adds the adaptive gradient using a multiplicative term in the denominator that sums the squares of previous gradients. This method allows the learning rate to be greater in the beginning of training where, presumably, the parameters are further from the minimum; and decreasing as the minimum is approached. However, a downside to this method is the possibility that the accumulated terms decrease the learning rate so much as to have slowed the learning too much before reaching the minimum. Subsequent developments have solved this issue.

Adadelta is a modification on the Adagrad algorithm in which it truncates the accumulation term at some length. This prevents the eventual suppression of the learning rate to 0. [cite paper]

Adam (for 'Adaptive Moment Estimation') is an algorithm which is a development from Adadelta and another algorithm called RMSprop. The Adam stores not only a average of the squared terms of past gradients—in Adam this is an exponential decay rather than a truncation, it also stores an average of past gradients. These are then used to update parameters similar to the above methods. [cite] Adam has become the dominant optimiser for neural networks within computer vision and natural language processing due to the result being favourable in comparison to other optimisers.

However, adaptive learning rate optimisers have the downside of not converging in certain circumstances. The details reason can be found here [cite paper]. In 2018, [cite paper] has modified the Adam algorithm where instead of using an exponentially decaying average, it uses the maximum value of past gradients to determine the learning rate and the momentum. This algorithm is called AMS-Grad and the authors have included a proof of convergence that Adam lacks. This algorithm outperforms Adam on small datasets

while shows similar or worse performance on other datasets. Since this algorithm is very new, it is yet to be determined if this algorithm is superior to Adam.

3 METHOD

This section describes the method from which a functional neural network is yielded from the initial GOTO images.

3.1 Cutout Algorithm

3.2 FNN

3.3 CNN

3.3.1 CNN from scratch

3.3.2 CNN from studying MNIST

4 RESULTS

4.1 FNN

4.2 CNN

4.2.1 CNN from scratch

4.2.2 CNN from studying MNIST

5 DISCUSSION AND CONCLUSION

...

The report should focus on the work done during semester two. The report should be written so as to be understandable to a non-specialist physics graduate and should follow the standard structure of a scientific paper.

The report should be self-contained in the sense that it should not be necessary to have to re-read your report from semester I in order to understand what the project is about. The report should therefore begin with a good introduction that explains the context of the work that you have done for your project. It should explain what your project was about, what you intended to do, and why this is important / useful / interesting. This introduction should include the references that underpin the work. There will, therefore, be some overlap with the literature review from semester I.

The report will be assessed independently by your supervisor and another academic. The second academic will normally be the same one who assessed your report in semester I. This means that he/she will be someone who works in the same general area as the project topic (e.g. high-energy physics), but will *not* be familiar with the details of the project. Do not assume that the assessor should remember things from your Semester I report. The marks of the second assessor carry equal weight as those of your supervisor, and so it is important that you explain clearly the context of your project and discuss any specialist techniques that you have used in language that is comprehensible to a physicist who is not working in exactly the same research area.

The main criteria for assessment will be the coherence of the scientific argument, the quality of the results obtained and their analysis, and the level of understanding displayed. However, marks may be deducted for poor written English, which includes spelling (remember to use the 'spell check' facility on the word processor), syntax and punctuation. Allowance will be made for students whose first language is not English. You will also lose marks for poor scientific style concerning equations, references, figures, tables, etc.

The two assessors will award you a mark out of 50 for your report. The 50 marks are broken down into two categories:

Content Aims and objectives, description of relevant theories / experimental techniques, quality of results obtained, interpretation and analysis of results. **(30 marks)**

Clarity and presentation Organisation and structure, conciseness, quality of language, clarity of presentation, references, quality and appropriateness of diagrams and figures. **(20 marks)**

The marks of the two assessors will be averaged, and will count as 50% of the overall semester II mark. The other 50% of the semester II marks are awarded as follows:

25% Project attempt.
25% Oral examination (a.k.a. the '*viva*').

The overall semester II mark counts as 75% of the PHY480 module mark, with the other 25% being your semester I report.

Special note for 2020. The coronavirus pandemic makes the submission and assessment of lab-books impractical. You are therefore no longer required to submit your lab-book, and the 25 marks previously assigned for "project attempt and lab-book" will now be allocated entirely to the supervisor's project attempt mark.

6 PREPARATION OF THE REPORT

The report should be written in appropriate scientific language. You should avoid the use of jargon, but if you do use jargon or acronyms, they should be defined before use.

The project plan that you wrote for your semester I report should be attached verbatim as Appendix A. In the write up of your work in semester II, you should discuss how well you kept to the plan. If you departed significantly from the plan, you should explain why this was done.

Special guidance for 2020 reports. This year, it is probable that many of you will have had to depart from your original plan due to the coronavirus pandemic. It is also possible that your project was affected by the industrial action. In these cases, please explain how your project was affected, and what revised plan you followed as a consequence. This statement should be included as Appendix B. Please make a reference to this appendix when you discuss your project plan in the main body of your report. You can write as much as you like in the Appendix, as the appendices are not included in the word limit. If your project was not affected, please just write something like "My project was not significantly affected either by the industrial action or the coronavirus" in Appendix B. Note that you might still have changed your plan even if there had been no industrial action or coronavirus. These changes to your plan should be discussed in your report, just as in all previous years.

The total length of the report should not exceed about 10,000 words, excluding the appendices. You can have as many words as you need for your appendices. A quick way to estimate the word count of a LaTeX document is to create a PDF file, select and copy the relevant parts of the report, and then paste into a word processor that has a word counting facility (e.g. WORD).

Some of you may wish to refer back to material from the preliminary work section of your first semester report. This is permitted, but you need to make sure that the semester II report is self-contained, as explained in Section 1. One option is to summarize the results from Semester I in an Appendix called something like "Summary of results from preliminary work in Semester I", and refer to this Appendix as needed within the main body of your
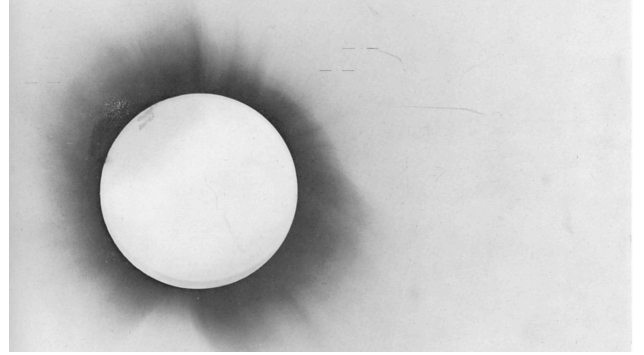
Figure 1. Solar eclipse of 1919, which was studied to prove the bending of light by gravity. From Dyson, et al. (1920).

report. If you include results from your Semester I report, you *must* make it clear that these results are from the previous report. For example, if you reproduce a data graph in the main body of your report, you must give a citation to your Semester I report, such as (Yours truly 2019). The reports will be submitted through Turnitin, which will pick up the overlap with the previous report. Unless the origin of the material from the previous report is properly acknowledged, its re-use is considered plagiarism, as that material has already been assessed.

The use of figures and tables can greatly improve the presentation of your project. For figures, the legends, annotations and axes labels, including units, must be clearly indicated. To include figures in a LaTeX document, you first need to generate an image file (JPEG, PNG or PDF), and then use the `includegraphics` command to include it in your document. Make sure that the axis labels are easily readable by ageing assessors with poor eyesight. Use the `figure` environment and the `caption` command to add a number and a caption to your figure. See the code for Figure 1 for an example. There is no limit to the number of figures you can include. Figures taken or adapted from published sources should be appropriately referenced, as in Fig. 1.

All equations should be numbered, as in eqn 6:

$$E = mc^2 . \tag{6}$$

The report must be submitted according to strict Journal formatting rules. See Section 8.

7 STRUCTURE OF THE REPORT

You are free to decide how best to present your work, but the following general structure should be used:

Front matter Title, registration number, date.
Abstract The abstract should be up to about 200 words long and summarize the work and its conclusions.
Introduction The introduction should describe the project aims and give an overview of the background to the research topic, explaining what other people have done before you.
Main body The main body of the report should begin with a discussion of any methods, theories, etc. relevant to the conduct of the project. You should then comment on your project plan. The results obtained and your analysis of them follows next. In some projects, it might be appropriate to summarize the key findings in a table. The final part is to discuss the results, putting them in their

context and highlighting their significance. The structure will thus typically be as follows:

- methodology / experimental techniques;
- discussion of any variations from your project plan submitted with your semester I report;
- results, and their analysis;
- critical discussion of results.

Conclusion The report should end with a conclusion that summarizes the results obtained, places the work in its proper context and, if appropriate, makes suggestions for further related work.

References A reference list should be included for all works that are cited in the report. See Section 9.

Appendices Appendix A should be the project plan from your semester I report. Appendix B should be a statement about the effects of the industrial action and coronavirus on your project. Other appendices (e.g. computer source code, additional experimental results) should then be labelled consecutively with letters starting at C.

8   REPORT FORMAT

The report should be submitted in one of two formats:

- Astrophysics projects should be submitted in the format of a paper in the *Monthly Notices of the Royal Astronomical Society*.
- All other reports should be submitted in the format of a paper in one of the *Physical Review* Journals. You should choose the one that is most appropriate to your branch of physics:

    – *Physical Review A* for atomic, molecular, and optical physics, and quantum information;
    – *Physical Review B* for condensed matter and materials physics;
    – *Physical Review C* for nuclear physics;
    – *Physical Review D* for particles, fields, gravitation, and cosmology;
    – *Physical Review E* for statistical, nonlinear, biological, and soft matter physics.

If you are in any doubt about which of the *Physical Review* formats to use, please ask your supervisor. The formats are, in fact, very similar, and you will not be penalized if you submit in the "wrong" *Physical Review* journal type.

The reports should be prepared following the instructions given on the journals' www sites.

- For *Monthly Notices*, please see https://academic.oup.com/mnras/pages/General_Instructions.
- For *Physical Review* instructions, please see: https://journals.aps.org/authors.

The easiest way to get the right format is to use the LATEX files provided by the Journal.

- For *Monthly Notices*, please see Section 2.1 of https://academic.oup.com/mnras/pages/General_Instructions.
- For *Physical Review*, you will need to use the REVTEX 4.2 package. Information about how to get the REVTEX 4.2 package is available at http://journals.aps.org/revtex/.

If you do use these LATEX files, your report will automatically be set in the right format. Hence we strongly recommend using LATEX for your report.

Some students may be unfamiliar with LATEX and prefer to use a word-processor like WORD to prepare the report. This approach is acceptable, but the Journals only provide LATEX templates, and you will have to fiddle around to make your report have the right format. Download a recent paper from the appropriate journal, and reproduce it using your word processor. It is not necessary to reproduce the journal-specific headers and footers, and you will not be penalized if your report is not exactly in the right format, provided it looks like a good approximation to a pre-print paper in the relevant journal, as in this template.

9   REFERENCES

You will need to cite the key papers that underpin your research in the introduction. You may need to give the references that explain the methodology that you are using, and you should compare your own results to those that are published in the literature.

In giving the references, you should use standard scientific style, providing the following information:

- author(s);
- title of paper;
- journal name, volume number, pages (or article ID number), year.

See Dyson, et al. (1920) or Einstein (1905) for examples. If you cite books (Fox 2010) or longer review articles, you should give the specific section of the book to which you are referring.

The references should be numbered for *Physical Review* style, or follow the Harvard (name, date) system for *Monthly Notices* format (as in this report template). LaTeX users might find it helpful to use BibTeX (BibTex 2020), while WORD users might want to consider using EndNote (Endnote 2020). If you are uncertain about any aspects of referencing, please look up the on-line tutorials provided by the library and refer to the Departmental referencing style for Physics & Astronomy (Library 1 2020).

10   SUBMITTING THE REPORT

You should submit your report electronically using the Turnitin anti-plagiarism portal of PHY480 by **Friday of week 12 of Semester II** (29th May). Please name your file: "XXXXXXXXX-PHY480-report", where "XXXXXXXXX" is your registration number. You should make sure that you comply stringently with the University's plagiarism guidelines (Library 2 2020). Note that minor changes to the original wording do not constitute a defence against accusations of plagiarism. If you are uncertain about any aspects of plagiarism, please look up the on-line tutorials provided by the library (Library 1 2020).

REFERENCES

BibTeX, 2020, see http://www.bibtex.org/

Dyson F. W., Eddington A. S. , and Davidson C. R., 1920, "Determination of the Deflection of Light by the Sun's Gravitational Field, from Observations Made at the Solar eclipse of May 29, 1919", Phil. Trans. Roy. Soc. A. 220, 291–333

Einstein A., 1905, "Zur Elektrodynamik bewegter Körper", Annalen der Physik, 17, 891–921.

Endnote, 2020, see https://www.sheffield.ac.uk/cics/endnote

Fox, M., 2010, *Optical properties of solids*, 2nd edn (Oxford University Press), pp 167 – 174.

Library 1, 2020, see https://www.sheffield.ac.uk/library/idlt/referencing
Library 2, 2020, see https://www.sheffield.ac.uk/ssid/exams/plagiarism
Student *registration number*, 2019, PHY480 Semester 1 Report, University of Sheffield

APPENDIX A: SEMESTER ONE PROJECT PLAN

Appendix A should be identical to the project plan from your semester I report.

APPENDIX B: EFFECT OF INDUSTRIAL ACTION AND CORONAVIRUS

Please give a statement in this appendix of the effect of the industrial action and coronavirus on your project. There is no word limit. If there was no significant effect, just write something like "The industrial action and coronavirus had no significant effect on my project."

APPENDIX C: OTHER STUFF

You can have as many other appendices as you need. These appendices could contain, for example, computer source code, additional experimental results, lengthy tables of data, etc. They could also contain material from your Semester I report to which you wish to make reference. These appendices should be labelled consecutively with letters, starting at C.

This paper has been typeset from a TeX/LaTeX file prepared by the author.