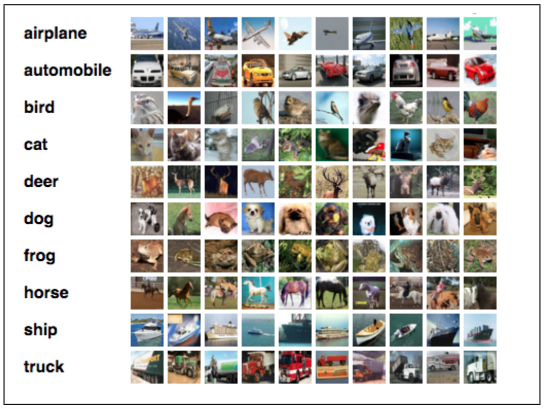
Coursework Assignment

Deadline: 13 May 2021, 4pm

Summary: You have to submit a **4-page report** (using the provided **template**) and an **implementation code**.

The assessment grade, which is worth 100% of the total grade, is separated into 2 components: the 4-page report and the source code.

The main task is to develop a deep neural network to perform multi-class classification. Specifically, the dataset for this assignment is a CIFAR-10 image dataset [1], which is also available from the torchvision library in PyTorch. The dataset is made up of 32x32 RGB images, which are split into a training set of 50,000 images and a test set of 10,000 images. The images have labels (one label per image) from 10 classes: *airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck*. The network is trained to predict the labels using the train data and its generalisation performance is evaluated using the test set.



TASK: Training a neural network

In this assignment, you need to implement a deep neural network with an input layer, **three hidden layers** (convolutional, fully-connected, or a combination of both) with ReLU non-linear activation, and an output (classification) layer. Feel free to use PyTorch (recommended), or any other deep learning frameworks (JAX, TensorFlow) [2] with automatic differentiation, or python. The neural network should be trained to classify the images from the CIFAR-10 dataset. You can use the built-in modules to load the dataset (e.g. in PyTorch, you can use torchvision.datasets.CIFAR10) and to build the layers in your model (e.g. in PyTorch you can use Linear, Dropout, ReLU, Conv2d, BatchNorm2d among others). The training process should explore the following hyperparameter settings:

- Batch size: Number of examples per training iteration.

[1] `http://www.cs.toronto.edu/~kriz/cifar.html`

[2] We recommend using Colab and enabling a GPU accelerator as in our lab sessions. You may also access a CPU server at the University of Sussex via the Citrix system.

- Depth: try and compare deeper versus shallower models. For example, compare performance when using two/three/four hidden layers.
- Width: Try using different numbers of hidden nodes and compare the performances. In a fully connected layer, this corresponds to the hidden layer size. In a convolutional layer, this corresponds to number of filters used for convolution.
- Convolutional filter size: try to vary filter size (also called the kernel size of the filters) in convolutional layers and compare the performance. Try to analyse, how the filter size affects the receptive field of the convolutional layers.
- Dropout: Dropout is an effective strategy to defend against overfitting in the fully connected layers. Try comparing the performance using different dropout rates.
- Batchnorm: batch normalisation is typically used in convolutional neural networks to prevent overfitting and speed up convergence. Compare performance with and without batch normalisation. Explore jointly with the batch size hyper-parameter.
- Max pool: max pooling is typically used to reduce the spatial dimensions of the layers (downsampling). Compare performance when using other types of pooling (average pooling), or no pooling.
- Tanh non-linearity: Compare the performance when training with tanh non-linear activation, with ReLU (this is the main task), and without any non-linearity.
- Optimiser: Try using different optimisers such as SGD, Adam, RMSProp.
- Weights initialisation: Try different weight initialisation strategies, such as He, Xavier, random noise.
- Regularisation (weight decay): L2 regularisation can be specified by setting the *weight_decay* parameter in optimiser. Try using different regularisation factors and check what effect this has on the performance.
- Learning rate, Learning rate scheduler: Learning rate is the key hyperparameter in model training; you can gradually decrease the learning rate to further improve your model. Try using different learning rates and learning rate schedulers to compare the performance.

You should explore the learning rate and at least 4 other types of hyperparameters (from those listed above); choose at least 3 different values for each hyperparameter (where applicable). For simplicity, you could analyse one hyperparameter at a time (i.e. fixing all others to some reasonable value), rather than performing a grid search.

You should describe your model selection procedure: 1) did you do a single train-val split or did you do cross validation, and how did you split the data? and 2) demonstrate your analysis of model selection based on learning curves: loss curves w.r.t. training epochs/iterations, accuracy curves on training and validation data. If you use TensorBoard to monitor your training, you can directly attach the screenshots of the training curves in your report.

To evaluate the performance of the model after hyper-parameter selection, you also need to have an evaluation part (for example a function), which uses (or loads) the trained model and evaluates its performance on the test set. In your report, please clearly state what hyperparameters you explored, how you did model selection, and what accuracy the model achieved on the train, validation and test sets.

Let your interest in designing, and deploying neural networks be a driving force when exploring hyper-parameter search in this assignment. We value your creativity and critical thinking when analysing and presenting your findings.

Details of Research Report

You are expected to write a 4-page report detailing your solution to the problem. Please use the provided latex or word template. Your report should include the following components (you are allowed to combine descriptions #2 and #3 but make sure we can easily identify them).

1. APPROACH (Maximum mark: 10) You should present a description of the neural network model you have adopted (e.g. did you use convolutional, fully connected layers, or a combination thereof; provide the details about layer sizes and any data pre-processing if required). Try to cover on a high level how the neural network works and notable assumptions on which the training approach depends. Pay attention to characteristics of the data set, for example, that CIFAR-10 contains visual data.

2. METHODOLOGY (Maximum mark: 30) Describe how you did training and testing of the neural network of your choice. This should include model selection (How did you split the data? which hyper-parameters were you selecting?), which optimisation approach did you use for training.

Describe any of your **creative solutions** with respect to overfitting and improving generalisation performance of the neural networks. Get inspired by the recently proposed strategies (described in the lecture 7 on model validation) such as cutout, and mixup. Reference to appropriate literature should be included.

3. RESULTS AND DISCUSSION (Maximum mark: 30) The main thing is to present the results sensibly and clearly. Present the results of your model selection. There are different ways this can be done:

- Use table or plots to show how the choice of hyper-parameters affect performance of the neural network using validation set (**refer to lectures on model validation, optimisation, convolutional neural networks**).
- Use graphs to show changing performance for different settings (**learning curves of the loss and train/val performance; refer to lectures on model selection, and optimisation**), if you choose to do that.

If any, provide analysis on the usefulness of taking into account advanced regularisation strategies to avoid overfitting and to improve generalisation.

You should also take the opportunity to discuss any ways you can think of to improve the work you have done. If you think that there are ways of getting better performance, then explain how. If you feel that you could have done a better job of evaluation, then explain how. What lessons, if any have been learnt? Were your goals achieved? Is there anything you now think you should have done differently?

Details of Code

You must also submit your implementation codes. Please make sure we will be able to run your code as is. High quality codes with a good structure and comments will be marked favourably. **Maximum mark: 30**

Marking Criteria

70% – 100% **Excellent**

Shows very good understanding supported by evidence that the student has extrapolated from what was taught, **through extra study or creative thought** (e.g. incorporating additional regularisation or data augmentation strategies). Work at the top end of this range is of exceptional quality. Report will be excellently structured, with proper references and proper discussion of existing relevant work. The report will be neatly presented, interesting and clear with a disinterested critique of what is good and bad about approach taken and thoughts about where to go next with such work.

60% – 69% **Good**

The work will be very competent in all respects. Work will evidence substantially correct and complete knowledge, though will not go beyond what was taught. Report should be well-structured and presented with proper referencing and some discussion/critical evaluation. Presentation will generally be of a high standard, with clear written style and some discussion of related work.

50% – 59% **Satisfactory**

Will be competent in most respects. There may be minor gaps in knowledge, but the work will show a reasonable understanding of fundamental concepts. Report will be generally well-structured and presented with references, though may lack depth, appropriate critical discussion or discussion of further developments, etc.

40% – 49% **Borderline**

The work will have some significant gaps in knowledge but will show some understanding of fundamental concepts. Report should cover the fundamentals but may not cover some aspects of the work in sufficient detail. The work may not be organised in the most logical way and presentation may not be always be appropriate. There will be little or no critical evaluation or discussion. References may be missing, etc.

30% – 39% **Fail**

The work will show inadequate knowledge of the subject. The work is seriously flawed, displaying major lack of understanding, irrelevance or incoherence. Report badly organised and incomplete, possibly containing irrelevant material. May have missing sections, no discussion, etc.

Below 30% **unacceptable (or not submitted)**

Work is either not submitted or, if submitted, so seriously flawed that it does not constitute a bona-fide report/script.