# Assessment 2: Details and instructions

Completion requirements

|  |  |
| --- | --- |
| **Assessment 2** | Working code and written report |
| **Due date** | 11.59 pm (AEST), Sunday 14 September 2025 (Week 7)  \*For current Melbourne time, please check information under the Assessment tile in the LMS of this subject. |
| **Weighting** | 30% |
| **Word count/length** | Working code and >1000-word report (+/– 10%; the reference list is not included in the word count but in-text citations are) |
| **SILOs** | * Implement a neural network with different learning algorithms for time-series forecasting with real-world data from industry (SILO 4). |

## Purpose

The purpose of this assessment is to develop hands-on experience with neural networks for image classification – a key application of machine learning used across industries such as health care, autonomous systems and digital security to interpret and act on visual data.

## Task details

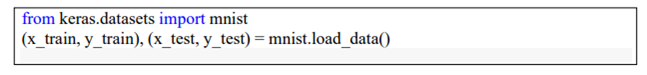
This assessment aims to consolidate your knowledge and practical skills to build neural networks (NNs) for supervised learning. The task is formulated as a multi-class classification problem for handwritten images, and the goal is to model the relationship between the images’ content, network structure and labels. You need to provide:

* working code (part 1)
* a written report of 1000 words on the method and results (part 2).

## Instructions

The **MNIST** database is a dataset with handwritten digits (from 0 to 9). The digits have been size-normalised and centred in a fixed-size image (28 × 28 pixels) with values from 0 to 1. You can use the following code with TensorFlow in Python to download the data.

**Figure 1**  
*Code to download the data*



 © La Trobe University, 2024

Every MNIST data point has two parts: an image of a handwritten digit and a corresponding label. We will call the images 𝑥 and the labels 𝑦. Both the training set and test set contain *𝑥* and *𝑦*.

Each image is 28 pixels by 28 pixels.

As mentioned, the corresponding labels in the MNIST are numbers between 0 and 9, describing which digit a given image represents. In this assessment, we regard the labels as **one-hot vectors**; that is, 0 in most dimensions, and 1 in a single dimension. In this case, the 𝑛-th digit will be represented as a vector, which is 1 in the *𝑛* dimensions. For example, 3 would be [0,0,0,1,0,0,0,0,0,0].

The assessment aims to build NNs for classifying handwritten digits in the MNIST database, train it on the training set and test it on the test set. Since the main object of this assessment is for you to understand the relationship between input, model and output, you are not expected to achieve very high accuracy in model performance; instead, for each task, you should be able to identify how you can improve model performance with the change of network structure.

There are two parts to this assessment.

## Part 1

Part 1 is comprised of three main tasks.

**Task 1**

Build a neural network without convolutional layers to do the classification task (hint: you will need the use of dense layers). Then you can change the model structure (i.e. number of dense layers, number of neurons in dense layers or activation functions) to be able to improve network performance.

**Task 2**

Build a neural network with the use of convolutional layers (you can decide other layer types you want to include in your network). Then you can change the number of convolutional layers and the number of filters or activation functions in the convolutional layers to be able to improve network performance.

**Task 3**

Change the type of optimiser or learning rate that you applied in the previous tasks and see how these changes can influence model performance. (You can keep the final network structure you applied in task 2 and try at least one different optimiser setting.)

Please read the following comments and requirements very carefully before starting the assessment:

1. The assessment is based on the content of labs and Weeks 1–3.
2. In Week 1 we talked about the use of training set, validation set and test set in machine learning. In this assessment, you are asked to train the NN on the training set and test the NN on the test set, without any given validation set. **(If you want to monitor the training process, you can also try what we did in Week 3: you can consider the validation set is the same as the test set in this assessment.)**

**No, I have still followed the conventional rules, and taken 10% of the training set as validation set.**

**The total sample size is 70,000. 60,000 are reserved for training, and 10,000 for testing. When we split the training sample further in 54,000 for training, and 6,000 for validation, we obtain a**

**Train | Val | Test ratio of**

1. In the assessment, the performance of an NN is measured by its prediction accuracy in classifying images from the test set (i.e. number of the correctly predicted images/number of the images in the test set).
2. Since the MNIST dataset is a black-and-white image dataset, the shape of dataset is (dataset\_length, 28,28). But to fit it into a conv2d layer, we need to make the input shape comply with its required format: (batch\_size, image\_width, image\_depth, image\_channels). Although batch\_size can be decided later when you train it, you will still need to tell the number of image channels here. You can consider reshaping the dataset into (dataset\_length, 28,28,1) or add one more dimension at the end with np.newaxis.
3. You are expected to show at least two models in for tasks 1 and 2: one for the model you start with, and another model is the model that you identified to have better accuracy. For task 3, you need to show what optimiser and/or learning rate you applied.

## Part 2

Your report **must** at least contain the following content:

1. Your name and student number.
2. Architectures of the NNs, with figures for tasks 1 and 2.
3. Description on the optimiser and learning rate you applied in the final model of task 2 and the optimiser or change of learning rate you used in task 3.
4. Experiments and performances, with parameter setting.
5. Discussion on the improvement/deterioration of the NN’s performance after changing the architecture and parameter setting for each task and findings of comparing the results from all three tasks.
6. The ranking of all NNs’ performances from all the three tasks.

## Assessment criteria

This assessment will measure your ability to:

**Part 1:**

* describe the two models, experiment settings and compare the results for task 1 (25%)
* describe the two models, experiment settings and compare the results for task 2 (25%)
* describe the two optimisers or learning rates, experiment settings and compare the results for task 3 (35%)

**Part 2:**

* demonstrate correct code quality (10%)
* research extensively and demonstrate depth of thinking; produce a well-structured report (5%).

Refer to the [marking guide](https://lms.latrobe.edu.au/mod/resource/view.php?id=8409518) for marking and feedback information.

## Submission details

The submitted assessment consists of (1) a report (in PDF format) of no less than 1000 words and (2) all codes for modelling, training and testing the NN with TensorFlow in Python (you can choose to have one code file including all your codes, or you can have one code file for each task separately).

**If you use ChatGPT or other generative AI tools, you must cite them and clearly indicate your original contributions.**

In keeping with La Trobe University policy, all assignments are to be submitted in Moodle via Turnitin.

To be accepted, your assessment submission **must** generate a similarity score (you are responsible for checking this). Submitting in Word or PDF format is the best way to do this. If your submission does not generate a similarity score, it cannot be checked for plagiarism and therefore **will not be marked.**

Last modified: Tuesday, 12 August 2025, 8:58 PM