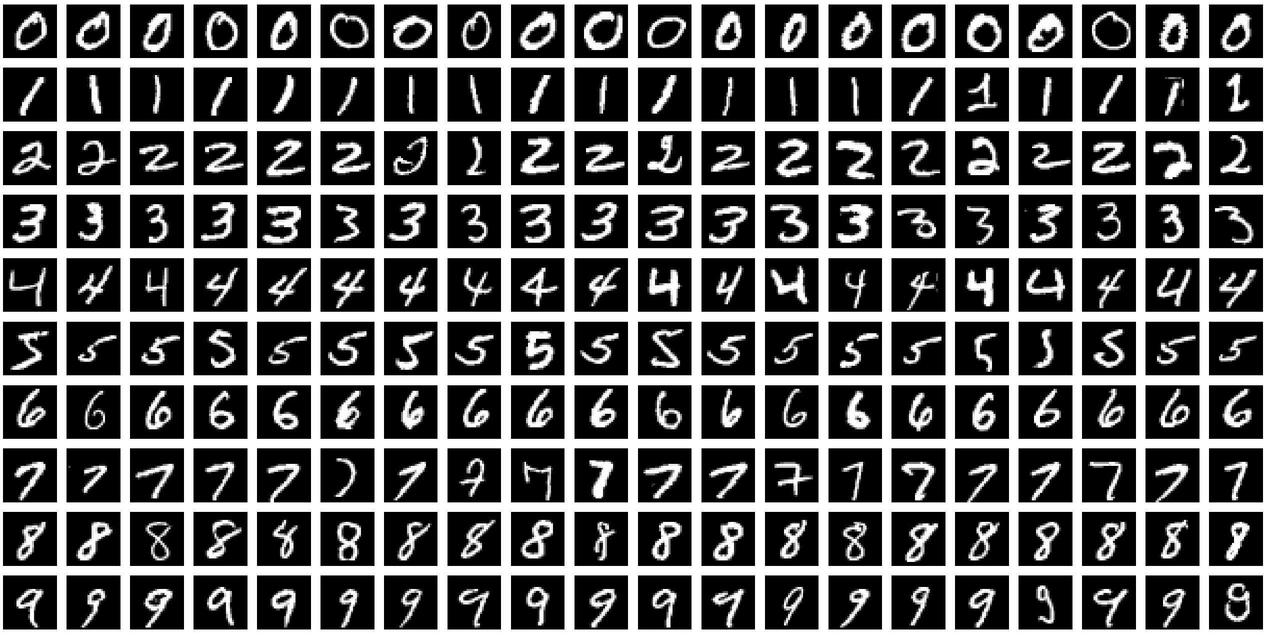
Problem 2: Neural Network with TensorFlow

1. Description

In this project, you will use TensorFlow – one of the state-of-the-art frameworks for deep learning – to solve a simple but real problem: recognizing hand-written digits. The MNIST dataset was created for this purpose in 1994 and has since served as a benchmark for evaluating machine learning models on image classification tasks. It consists of 70,000 grayscale images of handwritten digits (0-9), each of size 28x28 pixels. Your task is to develop a neural network which is trained on the MNIST dataset and achieves a state-of-the-art performance on the test set.



Source: Suvanjanprasai, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=156115980

You are provided with a Python environment[1] and a set of Python files which act as starting point. You must keep the overall structure of the project with the following four modules[2]:

- data.py – Loads the MNIST dataset (either training or test set) and pre-processes it.
- model.py – Implements the neural network model.
- train.py – Runs the entire training loop and validation.
- test.py – Loads a trained model and calculates the test metrics.

[1] Provided as a Dockerfile, which is pre-installed on the lab computers
[2] For your own projects – e.g. your Bachelor thesis or project 3 of the deep learning course – we highly recommend sticking to the same or a similar structure.

Follow these steps to implement the neural network:

1) Create the data loading and pre-processing class in data.py. You must create a tf.data[3] pipeline.
2) Create the neural network in model.py with the Sequential() function[4].
3) Create the training code with built-in fit() function[5] and verify that the end-to-end training process works.
4) Implement the testing code which calculates the test accuracy in test.py.
5) Replace the Sequential() model by a custom model that is built by creating a custom sub-class of the keras.Model class[6].
6) Replace the fit() function by a custom training loop that you build from scratch[7].
7) Extend your training loop with TensorBoard[8] functionality. You must log at least the training and validation losses, accuracies, the graph of your neural network, the histograms of all weights, and images of 12 misclassified images from the validation set.
8) Improve your neural network to reach state-of-the-art performance (> 90% accuracy).

Note: Steps 2) and 3) are intermediate steps which should help you to quickly get a running version which you can build the further steps on. However, you may skip these steps if you wish. The Sequential() and fit() functions may be very useful for simple problems, however for any advanced deep learning projects, their functionalities are usually too limited.

2. Formal requirements

Your submission on Moodle must contain the following files:

- data.py – which includes a working tf.data pipeline to load and pre-process the MNIST data.
- model.py – which includes a custom sub-class of keras.Model that defines you model (without calling the Sequential() function!).
- train.py – which implements a custom training loop (without calling the fit() function!) to train the model and which logs the information specified in section 1 to TensorBoard.
- test.py – which loads a trained model and calculates the test metrics.

You must also upload four print-screens of TensorBoard, showing the following TensorBoard tabs (which contain the information specified in task 7) of section 1.:

- Scalars
- Images
- Graphs
- Histograms

In addition, you must submit the final test accuracy in % in the corresponding text field.

[3] See https://www.tensorflow.org/guide/data
[4] See https://www.tensorflow.org/guide/keras/sequential_model
[5] See https://www.tensorflow.org/tutorials/quickstart/beginner
[6] See https://www.tensorflow.org/guide/keras/making_new_layers_and_models_via_subclassing#the_model_class
[7] See https://www.tensorflow.org/guide/keras/writing_a_training_loop_from_scratch
[8] See https://www.tensorflow.org/tensorboard/get_started

3. Submission & deadlines

Deadline for submission:	Monday, 28 April 2025 until 23:59
Submission via:	Moodle – DL FS2025 – Submission of Problem 2 See formal requirements on what must be submitted.

4. Evaluation

Your solution is marked with “pass” or “fail”. It is graded as “pass”, if

- Your solution was submitted on time.
- At least **5 out of 6 points** in the following evaluation is reached:

Criteria	0 Points	1 Point	Comment
1. Correct implementation of tf.data pipeline	☐	☐	
2. Correct implementation of custom model as a sub-class of keras.Model	☐	☐	
3. Correct implementation of custom training loop	☐	☐	
4. Correct implementation of TensorBoard functionality	☐	☐	
5. Test-accuracy of > 90% is reached	☐	☐	
6. Formal requirements are met	☐	☐	
Total points:	___ / 6 Points		
Pass / Fail:	☐ Pass ☐ Fail		