Assignment 6

TDT4171 — Artificial Intelligence Methods

March 2025

Information

- **Delivery deadline: March 6, 2025 by 23:59**. No late delivery will be graded! Deadline extensions will only be considered for extraordinary situations such as family or health-related circumstances. These circumstances must be documented, e.g., with a doctor's note ("legeerklæring"). Having a lot of work in other classes is not a legitimate excuse for late delivery.
- Cribbing("koking") from other students is not accepted, and if detected, will lead to immediate failure of the course. The consequence will apply to both the source and the one cribbing.
- Students can **not** work in groups. Each student can only submit a solution individually.
- Required reading for this assignment: Chapter 19. Learning from Examples (the parts in the curriculum found on Blackboard "Sources and syllabus" → "Preliminary syllabus") of Artificial Intelligence: A Modern Approach, Global Edition, 4th edition, Russell & Norvig
- For help and questions related to the assignment, **ask the student assistants during the guidance hours**. The timetable for guidance hours can be found under "Assignments" on Blackboard. For other inquires, an email can be sent to tdt4171@idi.ntnu.no
- Deliver your solution on Blackboard. Please upload your assignment as one PDF report and one source file containing the code (i.e., one .py file) as shown in Figure 1.

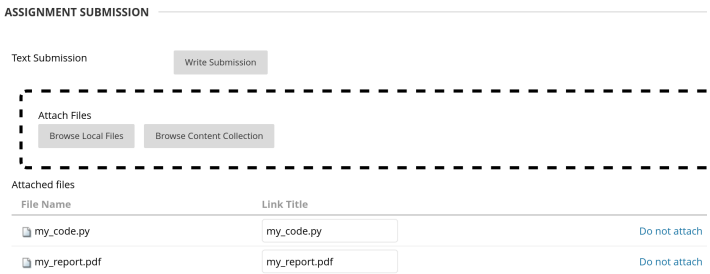

Figure 1: Delivery Example

Assignment Information

In this assignment, you will try out a fully connected neural network and a convolutional neural network on the MNIST-dataset, containing images of handwritten digits. The goal of the assignment is to get familiar with Keras and TensorFlow: We will use the Python package Keras. If you are unfamiliar with how to install packages in Python, you may want to take a look here.

TensorFlow is a library for deep learning. While flexible and powerful, it can be seen as having a steep learning curve. We will therefore use Keras as a simplifying abstraction layer in this assignment. Keras is part of TensorFlow, and thus TensorFlow needs to be installed. The installation guide for TensorFlow is available from here, and the getting started guide for Keras is available from here.

A Python file (assignment_6.py) is available at Blackboard. This contains the code you will need for this assignment. This code and the instructions given in this document has been tested on Python 3.11, Tensorflow 2.18.

About the data

The MNIST dataset contains 28x28 grayscale images of handwritten digits. Each image belong to one of 10 classes, that is the digits 0-9. There are 60 000 training images and 10 000 test images in the dataset. The function `load_data()` in the provided python file loads and preprocesses the dataset for model training. See Figure 2 for some example images from the dataset, marked with class label.

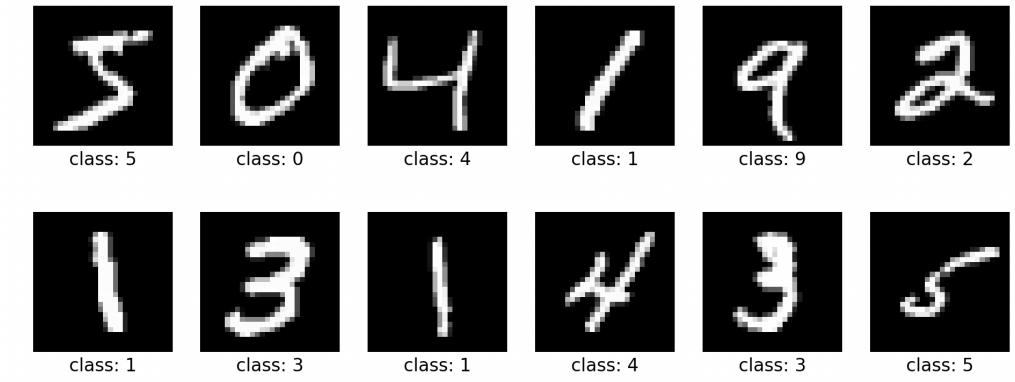

Figure 2: MNIST samples

Exercise 1

In the attached file assignment_6.py you will find code for building a deep neural network and for training the model on the MNIST dataset. The code should be runnable with no modifications.

The task is to get familiar with the code, and make the necessary changes in order to get the models to reach an acceptable accuracy on test data.

You will work with two types of models - a fully connected neural network/multilayer perceptron (MLP) and a convolutional neural network (CNN). The provided function `build_model()` returns returns an MLP for argument `cnn=False`, and it returns an CNN with `cnn=True`. Make sure to try both.

MLP

You will find a fully connected hidden layer of the MLP implemented in line 33 in the code. To increase the complexity of the model you can experiment with adding more units to the layer. You can also add more dense layers to the model to increase the depth.

Required test accuracy for the MLP is $> 90\%$.

CNN

You will find a convolutional layer and a pooling layer for the CNN on line 40-42. To increase the complexity of the model you can experiment with adding more filters to the convolutional layer and increasing the size of the kernel. You can also add more layers to the model.

Required test accuracy for the CNN is $> 96\%$.

The line `model.summary()` will print some information about the current model. Pay attention in particular to the total number of parameters displayed here.

In addition to model architecture, you should also consider changing the number of epochs for training and the learning rate passed to the optimizer.

(OPTIONAL: While not required for this assignment, feel free to experiment with other aspects of the model and training, such as type of optimizer, layer activations etc.)

The PDF report should include:

- Document the best accuracy reached on test-data for both model types (Include numbers in the PDF and deliver the modified code).
- After trying a few different model architectures and training settings, for both the MLP and the CNN, write a few sentences comparing the performance achieved for both models. Relate this to the design of the convolutional layer and the total number of parameters for each model.
- The model output is a softmax layer with 10 units, and the models are trained with the categorical cross entropy loss. Give a short explanation of how categorical cross entropy on softmax output learns to classify the MNIST-data.