Assignment 7

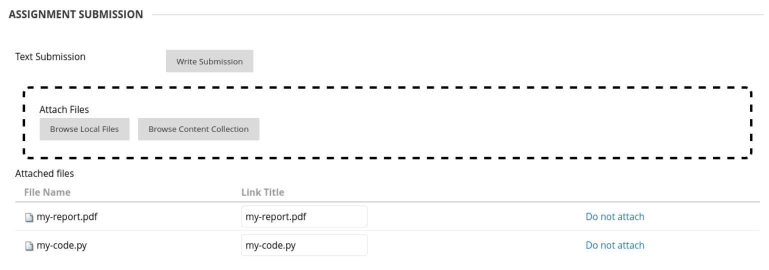
TDT4171 — Artificial Intelligence Methods

March 2023

Information

- **Delivery deadline: March 13, 2023 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 19.6 Linear Regression and Classification
  - Chapter 22.1 Simple Feedforward Networks
  - Chapter 22.2 Computation Graphs for Deep Learning

  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 "Course work" 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.



File Name	Link Title	
my-report.pdf	my-report.pdf	Do not attach
my-code.py	my-code.py	Do not attach

Figure 1: Delivery Example

1 Feedforward Neural Network

In this assignment, you will implement a feedforward neural network with one hidden layer that supports regression, like the neural network shown in Figure 22.3 in the textbook [1]. The neural network needs to take two input features. The hidden layer needs to support the sigmoid activation function (see Figure 22.2a) with two units. Finally, the output layer with one unit. Unlike the decision tree from Assignment 6, a neural network is a parametric model meaning that it learns from data by optimizing some parameters $\vec{\boldsymbol{w}}$ guided by a loss function

$$\mathbb{E}[\vec{\boldsymbol{w}}] = \frac{1}{2}\sum_{d \in \mathcal{D}} (t_d - o_d)^2 \tag{1}$$

where $\mathcal{D}$ is a set of training examples, each in the form $d = \langle \boldsymbol{x}_d, t_d \rangle$. We will refer to this process as training. To train a neural network with one hidden layer, you will need to implement the gradient descent algorithm outlined in Equation (19.4) with the loss function in Equation (1) and its derivative derived in Section 22.1.2 Gradients and learning in the textbook.

A Python script is handed out along with this assignment. The script provides the function `get_data(n_train: int, n_test: int)` that takes in how many training and test samples we want to generate and returns a tuple of four NumPy arrays `X_train`, `y_train`, `X_test`, and `y_test`. We have already prespecified the number of training and test samples in the script. `X_train` and `y_train` are the training input and output samples. Correspondingly, `X_test` and `y_test` are the test input and output samples. The arrays are structured so that rows represent samples and columns represent features. `X_train` is of shape $(280, 2)$, meaning that there are 280 samples with two features. Correspondingly, `y_train` contains 280 real values, which are the regression targets.

Your task is to implement a feedforward neural network that takes two inputs, has two hidden units using sigmoid activation and one output unit with linear activation. This exact neural network is shown in Figure 22.3 in the textbook. Use `X_train` and `y_train` to train the neural network you have implemented.

In your report, write down the mean squared error you get on the training and test set using the trained model. Document the hyperparameters you used, such as the learning rate and how parameters are initialized.

Note

Using implementation of neural networks from machine learning and deep learning libraries such as Scikit-learn [2] and Keras [3] are not allowed (fully or partially). Furthermore, it is not allowed to copy any code from the internet. However, supporting libraries, such as NumPy [4] is necessary and allowed.

The code must be runnable without any modifications after delivery. Moreover, the code must be readable and contain explaining comments. This is especially important if the code does not work. Finally, do not archive your source file when delivering on *Blackboard*.

References

[1] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach, Global Edition, 4th Edition*. Pearson, 2021.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[3] François Chollet et al. Keras. `https://keras.io`, 2015.

[4] Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.