

Deep Learning – 1RT720

Report for Hand-in assignment 1

Student Y

January 20, 2025

1 Introduction

This document includes general instructions for assignments and a template for Hand-in assignment 1 in Section 2.

General

Always indicate the question number associated with your answer (e.g., “Exercise 1”). The report does not have to be readable as a stand-alone document. However, include a brief repetition of the assignment questions, or answer such that it is clear what is addressed in your text and the context of your statements and observations given (i.e., do not just answer “42”, but instead write out “The answer to the Ultimate Question of Life, the Universe, and Everything is 42.”[1]). Be short and concise in your answers. Make sure that every figure, table, and code has a caption and is referenced in your text. Do not forget to put your name in the report.

Code

When writing code, make isolated functions for the specific parts of your code. This will help you structure your code. Remember that if you have a function that calls another function, you need to include both of them or write what the other function does and returns. Always write **comments** at the beginning of *each* function indicating **what it does, what inputs it takes, and what it returns**. **Attach the complete code** both as an appendix to the report and as a separate zip file. In L^AT_EX you can for instance use the *minted* package to include code (Read more about minted [here](#)).

Results

When presenting results, include plots of the optimization process, quantitative scores for your models and other relevant stuff to show the performance of your models. An easy-to-use and well-documented library in Python is *Matplotlib* ([link](#)). For plots, **always** write a caption explaining what is shown in the plot, and also what can be observed/concluded from the plot. Recall to always have labels on the axes, and when suitable a legend (if multiple plots in the same figure). Make the plots of a suitable size to be visible (if printed). See Figure 1 for an example, code to produce the plot is available in instructions for Hand-in assignment 1.

2 Template for reporting Hand-in assignment 1

Exercise 1. Implement a multi-layer neural network

In exercise 1 we implement ...

Comment on any important steps and design choices you have made in your implementation that you want to highlight. A handful of sentences should be enough.

The full code is available in Appendix A.

Exercise 2a. Evaluate a linear model

Include the requested performance plot and comment on the result. Provide details on batch size and learning rate.

See Figure 1.

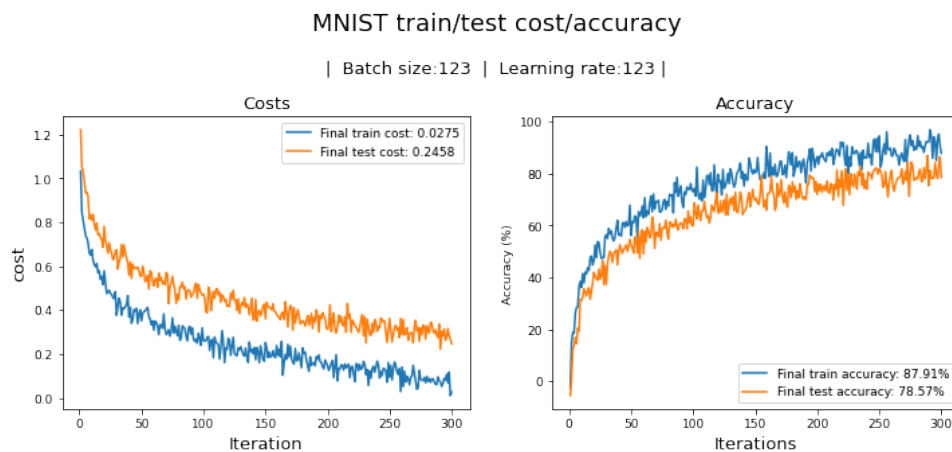


Figure 1: Plot showing how the cost and accuracy, evaluated on the training and test sets, is changing with number of iterations. It can be seen that... (note your curves will look different!)

Exercise 2b. Plot weight matrices

Include the requested plot of the weight matrices and comment on the result. How can we interpret the results? Note, some weight matrices are clearer to interpret than others, why?

As we can see in Figure 2...

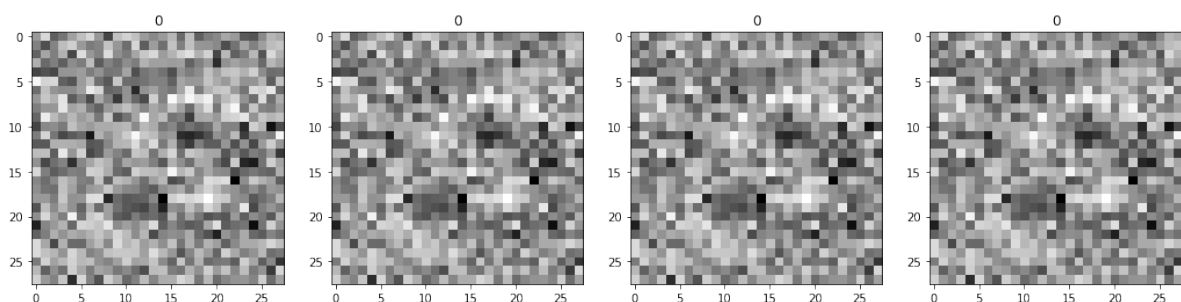


Figure 2: (your weight matrices will look different!)

Exercise 3. Evaluate your multi-layer neural network

Include the requested performance plot and comment on the result. Provide details on batch size, learning rate and model architecture (hidden units and activation function). Describe shortly how you came up with your design choices and if you tried other settings as well.

See Figure 3.

Figure 3: Plot showing the cost and accuracy for multi-layer neural network....

Use of generative AI

Write a few sentences if you have or have not used any generative AI, and if so how.

References

- [1] Adams, Douglas (1979). The Hitchhiker's Guide to the Galaxy. Pocket Books. p. 3.

A Code

```
1 def initialize(arg1, arg2, ...):
2     """ high level description of what the function does
3     param arg1: description of argument 1
4     param arg2: description of argument 2
5     ...
6     return: description of what the function returns
7     """
8
9     all_weights = ...
10    all_biases = ...
11
12
13    return all_weights, all_biases
14
15 def compute_cost(arg1, arg2, ...):
16     """ high level description of what the function does
17     param arg1: description of argument 1
18     param arg2: description of argument 2
19     ...
20     return: description of what the function returns
21     """
22
23    L =
24
25    return L
```
