Exercises week 44
FYS-STK3155 - Project 2

Ådne Rolstad
University of Oslo
(Dated: October 31, 2025)

In this project, we develop and implement a feed-forward neural network (FFNN) from scratch to investigate both regression and classification problems. Our main objective is to explore how neural networks can approximate complex functions and perform multiclass classification tasks, comparing the results with traditional machine learning methods such as ordinary least squares, ridge regression, and LASSO regression. For regression, we study the approximation of the analytical Runge function, while for classification we apply our model to the full MNIST dataset. We implement and analyze the backpropagation algorithm and gradient-based optimization methods to train the network efficiently. The results demonstrate that the FFNN achieves significantly improved predictive performance compared to linear models for non-linear problems, with good generalization when properly regularized. We also investigate the effects of network architecture, activation functions, and learning rate on convergence and accuracy. Overall, this project highlights the flexibility and power of neural networks for both regression and classification tasks when trained with well-tuned gradient descent methods.

I. INTRODUCTION

Machine learning has become a central methodology in modern science and technology, allowing us to uncover patterns in data and make predictions across a wide range of fields. Within this broad framework, neural networks stand out as particularly powerful models due to their ability to approximate complex, non-linear relationships and generalize across large datasets. Understanding how such networks work at a fundamental level is essential for building both theoretical insight and practical competence in modern data-driven modeling.

The main goal of this project is to develop a feed-forward neural network (FFNN) from scratch and use it to solve both regression and classification problems. In doing so, we aim to connect theoretical understanding with practical implementation. The project serves as a continuation of Project 1, where we explored linear regression and logistic regression methods. Here, we extend those ideas by implementing non-linear models capable of learning representations directly from data through the backpropagation algorithm and gradient-based optimization.

For the regression task, we investigate how neural networks can approximate analytical functions such as the Runge function and compare their performance with traditional methods like ordinary least squares (OLS) and ridge regression. For classification, we train our network on the MNIST dataset of handwritten digits, analyzing how model architecture, learning rate, and activation functions influence accuracy and generalization.

This report is organized as follows: Section II provides a theoretical overview of neural networks, including activation functions, cost functions, and the backpropagation algorithm. This section also describes the implementation details and methods used for training. Section III presents and discusses the results for both regression and classification. Finally, Section IV offers concluding remarks and suggestions for further work.

II. METHODS

A. Method 1/X

- Describe the methods and algorithms, including the motivation for using them and their applicability to the problem
- Derive central equations when appropriate, the text is the most important part, not the equations.

B. Implementation

- Explain how you implemented the methods and also say something about the structure of your algorithm and present very central parts of your code, not more than 10 lines
- You should plug in some calculations to demonstrate your code, such as selected runs used to validate and verify your results. A reader needs to understand that your code reproduces selected benchmarks and reproduces previous results, either numerical and/or well-known closed form expressions.

C. Use of AI tools

- Describe how AI tools like ChatGPT were used in the production of the code and report.

We have used ChatGPT for code review and debugging.

III. RESULTS AND DISCUSSION

Figure 1 shows how the test accuracy of a neural network depends on the number of hidden layers and the number of nodes in each layer. In this case, the cost function was cross-entropy and the activation function was ReLU in the hidden layers.

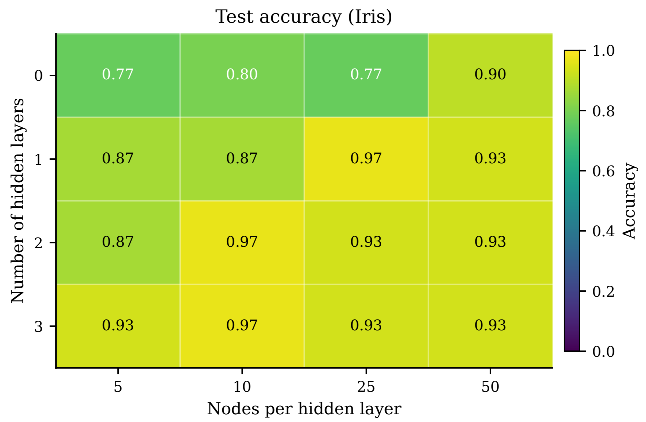

Figure 1: Heatmap showing test accuracy for neural networks with a varying number of layers and nodes per layer.

Figure 2 is an example of a "bad" figure. This figure is intentionally designed to demonstrate poor scientific visualization practices. The title is vague and does not describe what is being shown, while the axis labels ("x" and "y") provide no information about the variables or their units. The color choices and background reduce readability and contrast, making the data harder to interpret. The thick red dashed line, oversized markers, and inconsistent font sizes create visual clutter and distract from the actual results. In addition, the axis limits extend far beyond the data range, distorting the visual impression. Overall, the figure fails to communicate the underlying results clearly or professionally.

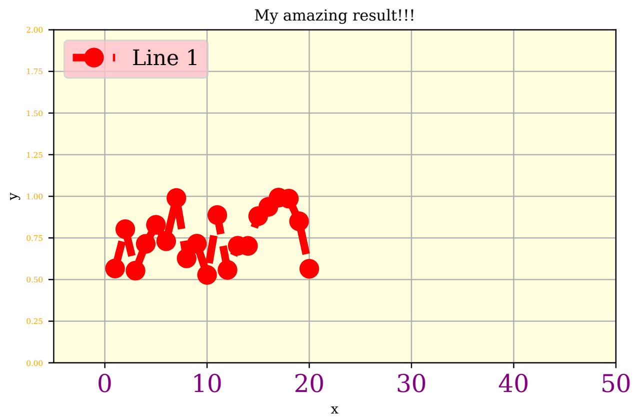

Figure 2: Bad figure, example

IV. CONCLUSION

- State your main findings and interpretations
- Try to discuss the pros and cons of the methods and possible improvements
- State limitations of the study
- Try as far as possible to present perspectives for future work