Report Template
FYS-STK3155 - Project 2

Student
University of Oslo
(Dated: October 31, 2025)

We investigate the performance of feed forward neural networks (FFNN's) on regression and classification tasks. For regression, we test a dataset taken from the one dimensional Runge function. Using k-fold cross validation with 5 folds, we test the mean squared error (MSE) for different numbers of layers and hidden nodes, gradient descent methods and activation functions. We also test the performance with l_1 and l_2 penalties with different regularization parameters. The results of the optimal neural network is compared with ordinary least squares (OLS) regression for a polynomial fit with degree 12. Next, we do a classification analysis on the multiclass MNIST dataset. We try different numbers of hidden nodes, activation functions and regularization parameters in order to minimize the softmax cross entropy cost. Our final results with both regression and classification are compared with implementations using PyTorch and Autograd. With this exploratory study, we have gained a better understanding of the behavior of neural networks, as opposed to simpler pre ML methods.

The abstract gives the reader a quick overview of what has been done and the most important results. Try to be to the point and state your main findings. It could be structured as follows:

- Short introduction to topic and why its important
- Introduce a challenge or unresolved issue with the topic (that you will try to solve)
- What have you done to solve this
- Main Results
- The implications

I. INTRODUCTION

When you write the introduction you should focus on the following aspects:

- Motivate the reader, the first part of the introduction gives always a motivation and tries to give the overarching ideas. Citing some central ideas or problems in the literature is a good idea here. [1][2][3, 4]
- What you have done, with a focus on choice of problem and method, and why these were chosen.
- The structure of the report, how it is organized. List the sections, and very briefly describe what is in them and how they fit together.

Neural networks have been an indespensable tool in the modern AI and machine learning revolution. They extend the usual methods of linear regression and binary logistic classification to allow deep learning through several internal layers connected by tunable weights and biases. Thus, a neural network can be trained to incrementally combine and identify more complex features in input data. One of the main types of neural networks is the fully connected feed forward neural network (FFNN). In a FFNN, information flows only in one direction through the network, and each node in one hidden layer is connected to all nodes in the next layer.

Each layer in a FFNN consists of a certain number of nodes or "neurons", each of which is associated with a certain value. The values in the nodes of the next layer is found by feeding the values in the current layer into an affine transformation (given by a multiplication of a weights matrix and a bias term) and a non-linear activation function. Traditionally, the activation function would be the step function, in order to mimic the behavior of real neurons in the brain. Nowadays, typical activation functions might be the ReLU family of functions, the sigmoid function and the tanh function. A neural network is trained by tuning the internal weights and biases of each affine transformation in order to reach a minimum of a chosen cost function. Typically, the parameters are tuned using a gradient descent (GD) algorithm. We then use the backpropagation to calculate the gradient of the cost function with respect to the parameters.

In this work, we explore how the FFNN's perform with different model architectures (given by the number of layers and nodes in each layer), training methods and activation functions. We also try with and without l_1 and l_2 regularization in the cost functions. We perform a regression analysis on data taken from the one dimensional Runge function, using the mean squared error (MSE) as cost function. We also perform a classification task on data from the MNIST dataset, using the cross entropy as cost function. For each choice of architecture, we plot the cost as a function of the learning rate used to train the model, using the Adam and RMSprop GD algorithms. The goal of our analysis is to become familiar with the performance of neural networks, including potential pitfalls with exploding or vanishing gradients in the cost function.

In section II, we explain the theory underlying neural networks and the backpropagation algorithm. We also briefly explain how we implemented them in Python. In section III, we present our main results and discuss them. In particular, we summarize and critically evaluate the various methods employed. Finally, we end in section IV with a conclusion and ideas for future work.

II. METHODS

A. Single-layer networks

B. Feed Forward Neural Networks

1. Forward pass

2. Back propagation

C. Activation functions

D. Cost functions

1. Regularization

- 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.

E. 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.

F. Use of AI tools

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

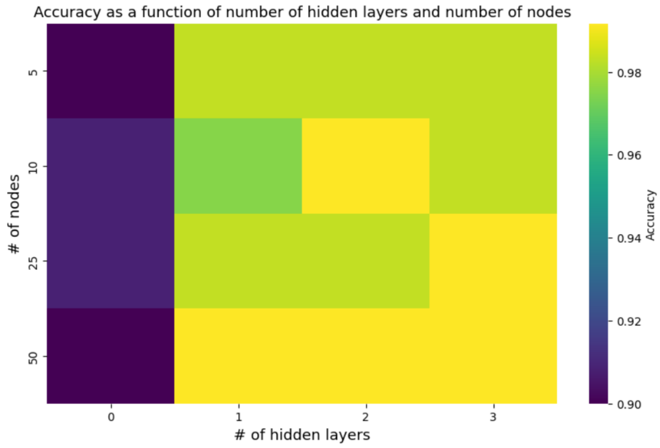

Figure 1: Test accuracy on test iris dataset after training on training iris dataset, as a function of the number of hidden layers and the number of nodes in each layer.

III. RESULTS AND DISCUSSION

In figure 1, we show the test accuracy on a testing datasubset from the iris data, after training on the training dataset. The training and testing data were found by splitting the iris dataset with a test size of 20%. We trained using Stochastic gradient descent, using RMSprop with a learning rate of 0.01. The training stopped after the cross entropy cost was within had a smaller difference between the maximum and minimum value than 10^{-4} in the last 20 epochs.

In figure 2, we show an example of a terrible figure that nonetheless contains axis labels, a title and a caption. The main problems with the figure are as follows: the title and axis labels are way too small to read. These should be roughly the same size as the text in the document. Second, the figure contains no colorbar, so one does not know what values the colors represent. Third, the caption and figure title (if it can be read at all) give us no useful information. They do not tell us anything about what quantity, exactly, we are seeing on the heatmap.

We refer to [5] for information about neural networks, gradient descent optimization and regression and classification. In [6], some challenges with neural network optimization are explained.

- Present your results

- Give a critical discussion of your work and place it in the correct context.

- Relate your work to other calculations/studies

- An eventual reader should be able to reproduce your calculations if she/he wants to do so. All input variables should be properly explained.

- Make sure that figures?? and tables contain enough information in their captions, axis labels etc. so

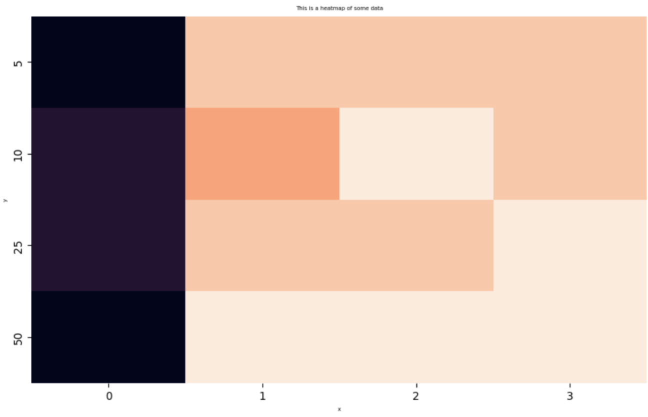

Figure 2: This plot shows some data.

that an eventual reader can gain a good impression of your work by studying figures and tables only.

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

[1] M. Hjorth-Jensen, *Computational Physics Lecture Notes 2015* (Department of Physics, University of Oslo, Norway, 2015), URL `https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf`.

[2] H. J. T. Zhang Yi, Yan Fu, Computers and Mathematics with Applications **47**, 1155 (2004), URL `https://doi.org/10.1016/S0898-1221(04)90110-1`.

[3] K. B. Hein, *Data Analysis and Machine Learning: Using Neural networks to solve ODEs and PDEs* (Department of Informatics, University of Oslo, Norway, 2018), URL `https://compphysics.github.io/MachineLearning/doc/pub/odenn/html/._odenn-bs000.html`.

[4] T. Hastie, R.Tibshirani, and J.Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer Series in Statistics* (Springer, New York, 2009), URL `https://link.springer.com/book/10.1007%2F978-0-387-84858-7`.

[5] M. Hjorth-Jensen, *Applied data analysis and machine learning* (2025), URL `https://compphysics.github.io/MachineLearning/doc/LectureNotes/_build/html/intro.html`.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016), `http://www.deeplearningbook.org`.