FNN placeholder

Kim Graatrud, Simon Silverstein, and Nicholas Andrés Tran Rodriguez

Universitetet i Oslo

Neural Networks are mathematical models designed to replicate the way brains learn. To do this we implement a 'neuron' which compute one feature with a weight and bias, this neuron alone cannot replicate a complicated function like Runge's function. We therefore scale it up to many neurons and show that many neurons can preform complicated regression and classification tasks.

CONTENTS

I. INTRODUCTION

Neural networks are mathematical models aiming to replicate or classify data using interconnected points called 'neurons', the structure of which is reminiscent how the brain wires neurons together to learn and recognize patterns. While one neuron may struggle to recognize any patterns, many of them may be capable to reproduce complex functions.

While there are several types of neural networks, each with their own use cases, we will in this paper focus on the uses of Fast Forward Neural Network (FFNN for short) for regression and classification.

Firstly we will aim to train our neural network on Runge's function for different number of hidden layers, as well as how it responds to changes in hyperparameters. We also aim to see how L1 and L2 regularization affects the learning of the network.

Then we will study how we can change the cost function and activation functions to make the network classify instead.

All code made for the production of the results for this paper can be found at our *GitHub* repository[1].

II. EXERSICE 2 ANSWERS

1. What is the main difference between ordinary least squares and Ridge regression?

   The main difference is that the on OLS the cost function is purely dependent on the relation between the dataset and the model as OLS is described as $(\tilde{y} - y)^2$. Ridge is described by the same cost function as OLS just with the L2 norm. This in turn makes it so that the the parameters that optimize the model to the dataset are also being minimized. Punishing each paramater W introduces an inherrent bias as now the trained model is not purely dependent on the data. The introduced bias varies by how large λ each parameter W are being punished. In other words the difference between OLS and Ridge is the inherent bias introduced to training the data by introducing a punishment term to each parameter.

2. Which kind of data set would you use logistic regression for? The dataset you would use logistic regression for is when working with a categorical dataset; such as a dataset of cancer patients and non cancer patients.

3. In linear regression you assume that your output is described by a continuous non-stochastic function . Which is the equivalent function in logistic regression? The equivalent function is.

$$f(x) = \frac{1}{1 + \exp(-x)}$$

4. Can you find an analytic solution to a logistic regression type of problem? No it is not possible as the gradient of the Log-Loss function contains non linear components $f(x)$ and logarithms that make it impossible to solve the logistic regression analytic.

5. What kind of cost function would you use in logistic regression?

It depends on how many categories you have. If you have two categories i.e cancer and not cancer you can use the binary cross-entropy cost function. If there are multiple categories you are training your model after you would use the multiclass cross-entropy cost function.

b) Deep learning

1. What is an activation function and discuss the use of an activation function? Explain three different types of activation functions?

   An activation function are non linear functions that treat connections between each hidden layer into

[1] https://github.com/KimGraatrud/FYS-STK4155-Projects/tree/main/Project_2

an output value. The activation function is necessarily non linear so that each layer does not do a linear transformation of each layer. Three examples of hidden layers are ReLu, Sigmoid function and Leaky ReLu.

2. Describe the architecture of a typical feed forward Neural Network (NN).

   The architecture is an x amount of input nodes. Thereafter you have y hidden layers with varying nodes.

3. You are using a deep neural network for a prediction task. After training your model, you notice that it is strongly overfitting the training set and that the performance on the test isn't good. What can you do to reduce overfitting?

   We can introduce a bias on each hidden layer.

4. How would you know if your model is suffering from the problem of exploding gradients?

5. Can you name and explain a few hyperparameters used for training a neural network?

6. Describe the architecture of a typical Convolutional Neural Network (CNN)

7. What is the vanishing gradient problem in Neural Networks and how to fix it?

8. When it comes to training an artificial neural network, what could the reason be for why the cost/loss doesn't decrease in a few epochs?

9. How does L1/L2 regularization affect a neural network?

10. What is(are) the advantage(s) of deep learning over traditional methods like linear regression or logistic regression?

c) Optimization part

1. Which is the basic mathematical root-finding method behind essentially all gradient descent approaches(stochastic and non-stochastic)?

2. And why don't we use it? Or stated differently, why do we introduce the learning rate as a parameter?

3. What might happen if you set the momentum hyperparameter too close to 1 (e.g., 0.9999) when using an optimizer for the learning rate?

4. Why should we use stochastic gradient descent instead of plain gradient descent?

5. Which parameters would you need to tune when use a stochastic gradient descent approach?

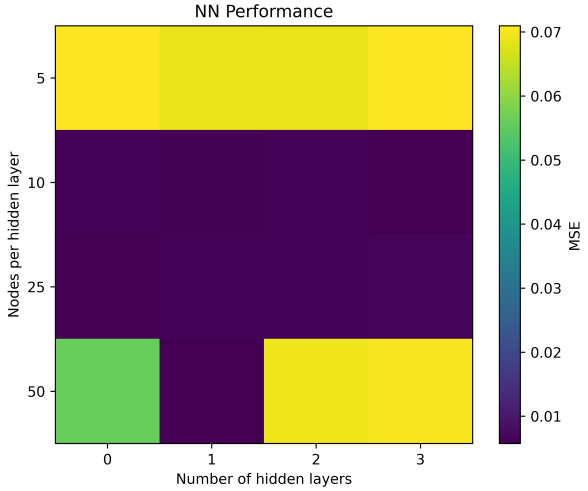

FIG. 1 Performance of FFNNs with different numbers of nodes and layers as trained on the 1-D Runge function. Color indicates the mean-squared-error from 100 predictions of test data. All networks were trained for 4000 epochs of batch size 50, with a learning rate of $\eta = 1.2$ and $a(z) = \sigma(z)$.

d) Analysis of results

1. How do you assess overfitting and underfitting?

2. Why do we divide the data in test and train and/or eventually validation sets?

3. Why would you use resampling methods in the data analysis? Mention some widely popular resampling methods.

4. Why might a model that does not overfit the data (maybe because there is a lot of data) perform worse when we add regularization?

Figure 2 has several bad qualities. Firstly, its lacking a overarching title, meaning we cant easily tell what the figure is supposed to show, making us rely on our intuition to understand what [1.] or [10, 1] refers to. Secondly, the graph is way too small and therefore hard to read any of the points or features. Its also not in a vectorized file format. The figure is missing a title, and not a single axis has an axis label. The caption of the figure doesn't properly describe the figure, and the caption should describe so much that it can be interpreted without the context of the paper.

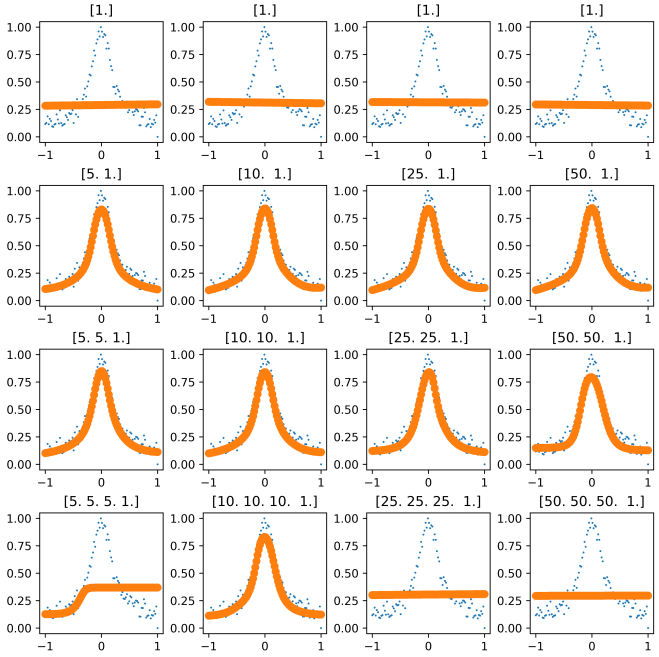

FIG. 2 Sources of the mean squared error.