Bachelor's degree in Aerospace Technology Engineering
The School of Industrial, Aerospace and Audiovisual Engineering of Terrassa

Numerical tools in machine learning for aeronautical engineering

Assignment: Neural Networks

Authors:
Francesc Antich Navarro
Lluc Ramon Busquets
Yi Qiang Ji Zhang
Alexis Leon Delgado

Professor:
Alex Ferrer Ferré

Contents

1 Introduction

Neural networks are a collection of algorithms that are roughly fashioned after the human brain and are meant to identify patterns. They analyze sensory data by categorizing or grouping raw input using machine perception. The patterns they identify are numerical, encoded in vectors, into which all real-world data must be transformed, whether pictures, music, text, or time series.

Neural networks assist us in clustering and classifying data. One can consider them a grouping and classification method above the data the user save and manage. They also hep to in the grouping of unlabeled data based on similarities between example inputs, and they classify data when they have a labeled dataset to train on.

Neural networks may also extract features that are fed into other algorithms for clustering and classification, thus deep neural networks may be thought of as components of broader machine-learning systems that include algorithms for reinforcement learning, classification, and regression.

2 Results

2.1 Classification solutions

In the following figures, we can see the boundary decision of each group. In both cases, we have used linear inputs and 2 layers, a lambda of 10^{-2} and 25% of test/training ratio. The program doesn't always find a good separation between the groups. We think this is because the fminunc function is not able to find the optimal minimum in all cases.

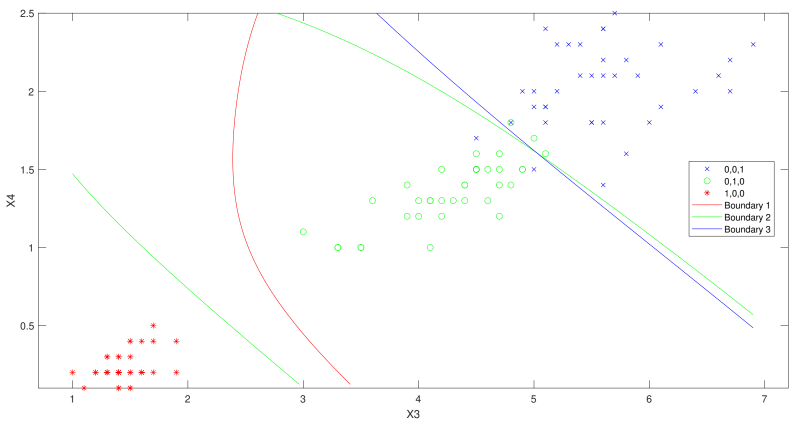

Figure 1: Boundary decision for 2 layers and $\lambda = 10^{-2}$

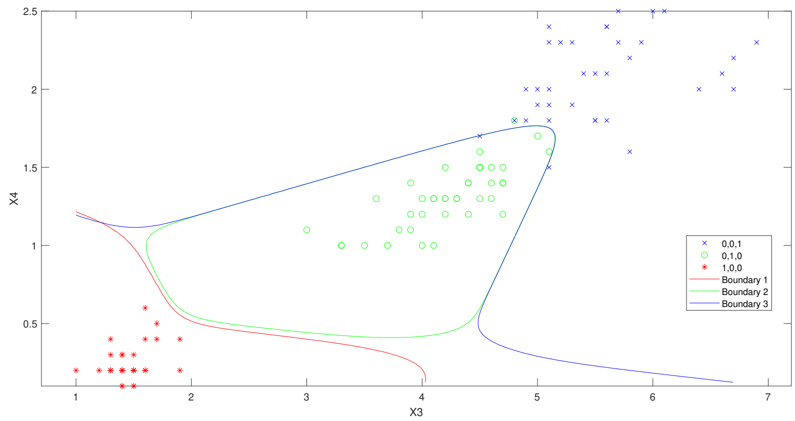

Figure 2: Boundary decision for 2 layers and $\lambda = 10^{-2}$

In figures 3 and 4 we can see the training error vs iterations of the fminunc function. In figure 3 the function has found a minimum with a function value of 0.2. In figure 4 the fminunc function hasn't found a minimum as good as the other case, the function value is over 0.8. Only the run of figure 3 has been able to plot a good boundary decision. We haven't been able to plot the generalization error vs iterations becasue we were not able to extract the values of theta for each iteration.

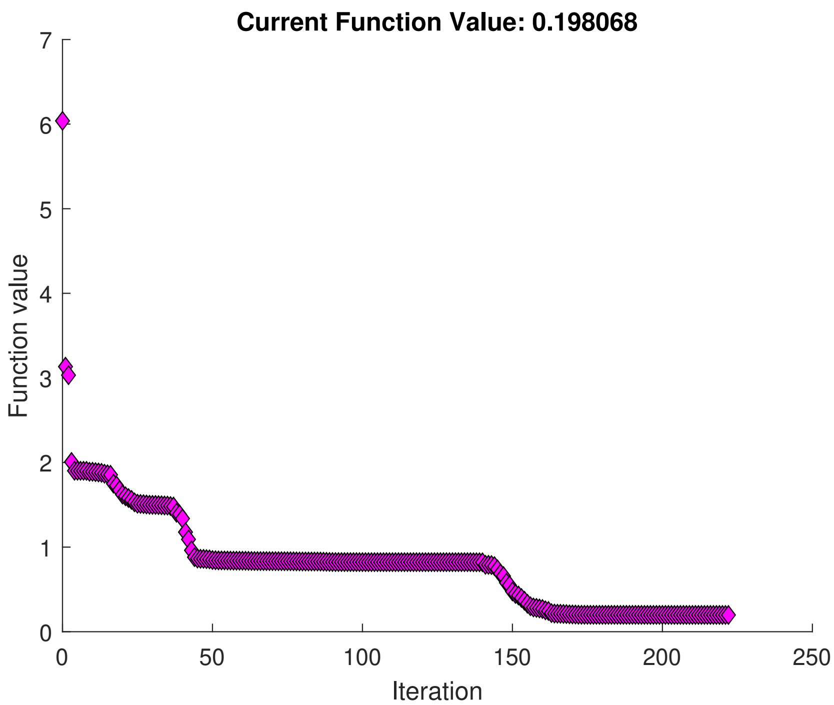

Figure 3: Training error vs iterations

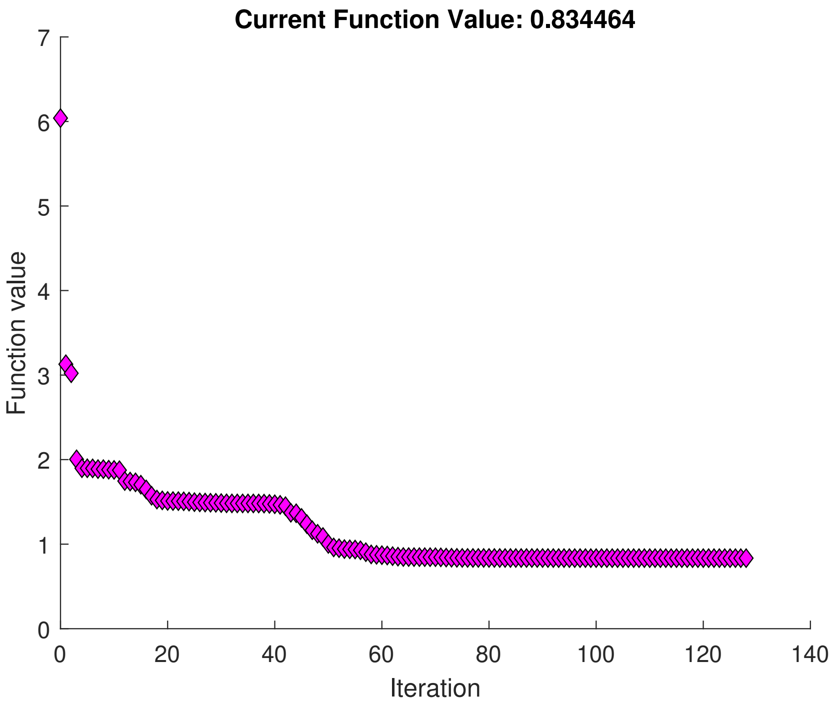

Figure 4: Training error vs iterations

2.2 Generalization error as function of number of layers

To study the dependence of the generalization error on the number of layers, a test/training ratio of 25% and a range of lambdas ranging from $\lambda = 10^{-1}$ to $\lambda = 10^{-5}$ has been chosen. In Figure 5, the progression of the generalization error vs the used number of layers can be observed.

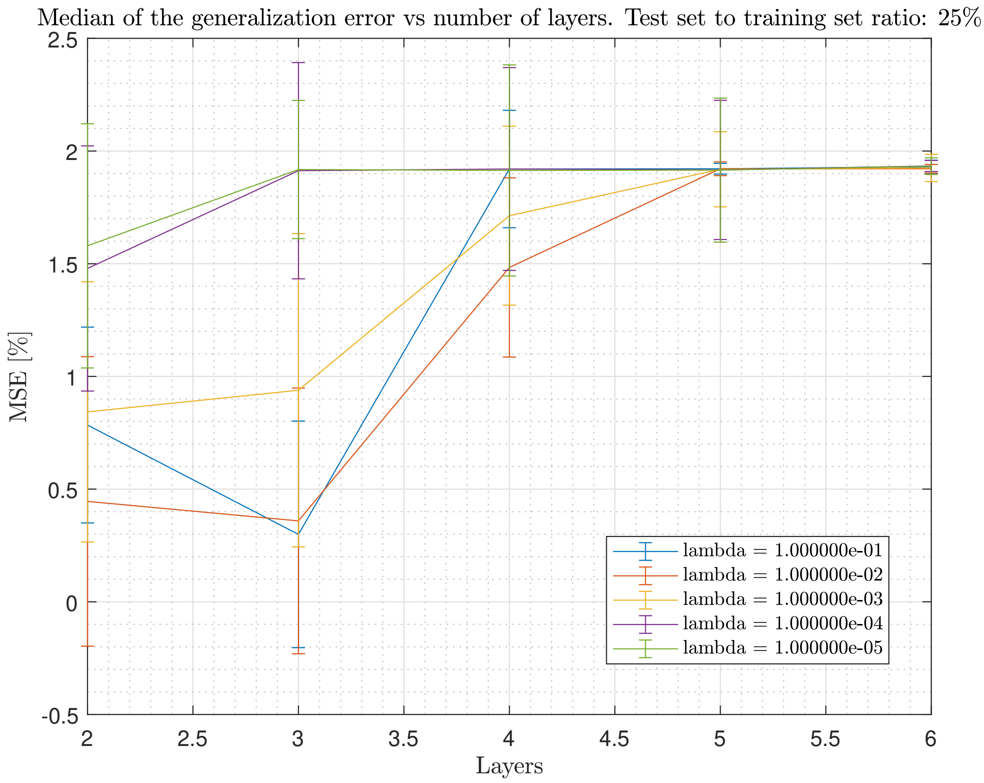

Figure 5: Generalization error (MSE) vs number of layers for a test/training ratio of 25%

It is worth noting that the usage of excessive number of layers, in this case ≥ 4, leads to a loss of information in the data due to its high level of processing among layers. That is to say, a lower number of layers must be used to represent properly the dataset information. In this case, three layers is the optimal value.

In addition, the greater the lambda is, the higher the optimal value of the number of layers used is.

2.3 MSE as function of λ

Figure 6 shows the impact of changing the regularisation parameter λ in the performance of the neural network once optimised. Note how as it approaches zero its effect gets neutralised, whilst for slightly greater values the test MSE gets reduced. The optimal value is found to be around 10^{-2}.

The percentage value is that of the test set, and the results are consistent with the fact that the fewer the training data, the greater the variance in the results obtained by the neural network.

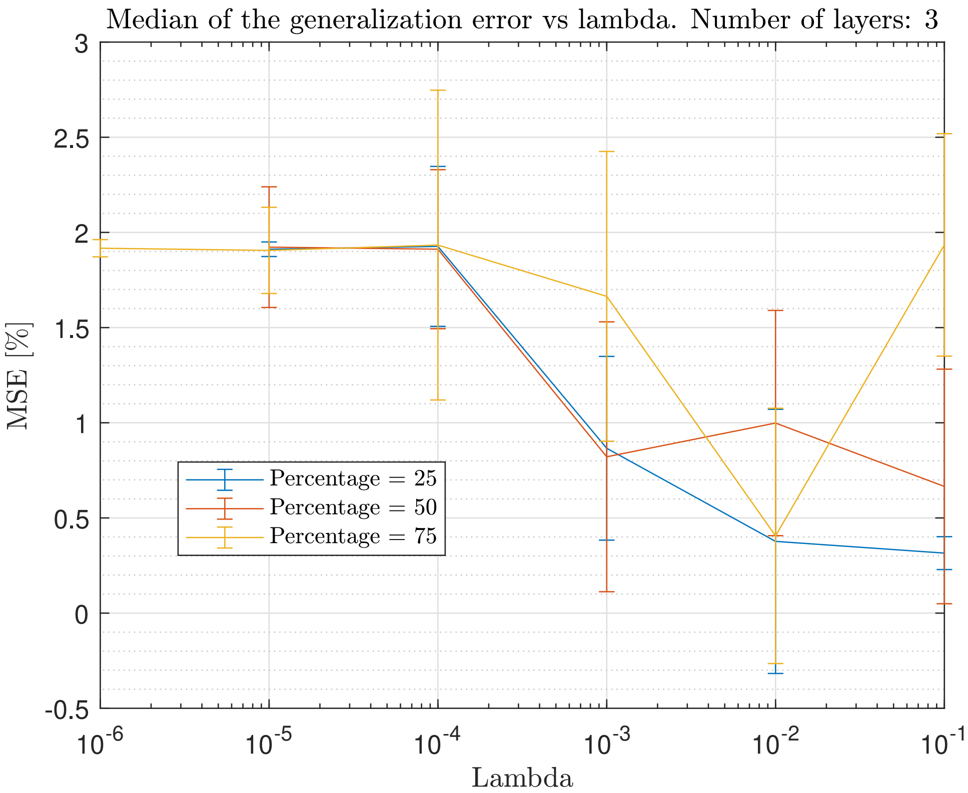

Figure 6: Generalization error (MSE) as a function of λ