**Artificial Neural Networks**

*Exercise Session 1 - Supervised learning and Generalisation,*

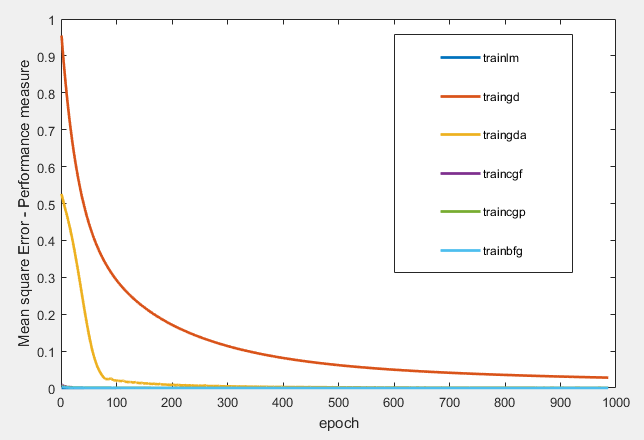
*report by,*

*Marimuthu Ananthavelu (r0652832), KU Leuven.*

***Comparison of various algorithms:***

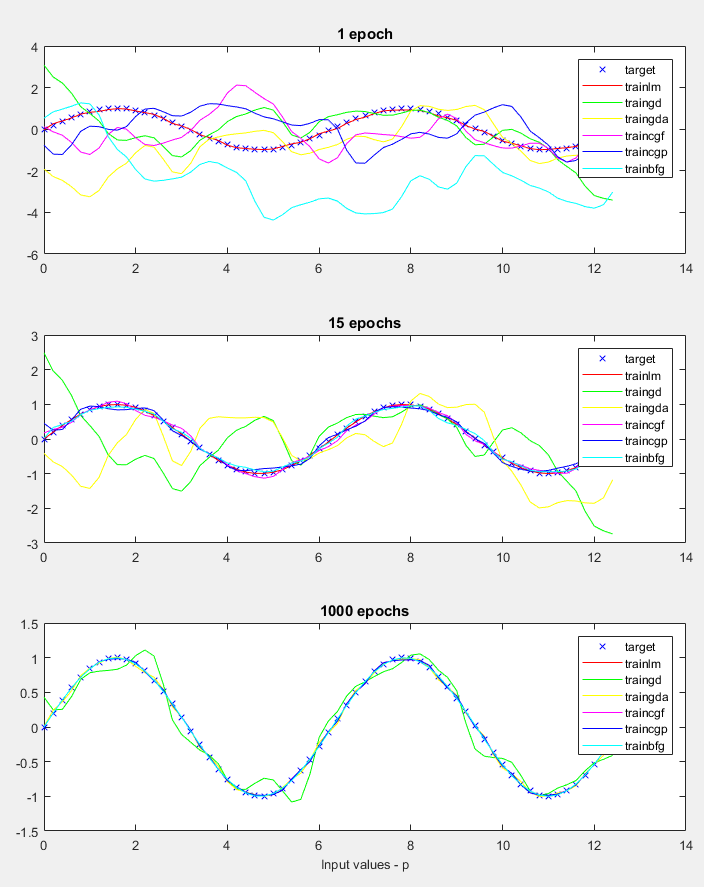
The following Backpropagation algorithms are compared for performance with the given data.

1. ***trainlm*** is a network training function that updates weight and bias values per Levenberg-Marquardt optimisation.
2. ***traingd*** is a network training function that updates weight and bias values per gradient descent.
3. ***traingda*** is a network training function that updates weight and bias values per gradient descent with adaptive learning rate.
4. ***traincgf*** is a network training function that updates weight and bias values per conjugate gradient backpropagation with Fletcher-Reeves updates.
5. ***traincgp*** is a network training function that updates weight and bias values per conjugate gradient backpropagation with Polak-Ribiére updates.
6. ***trainbfg*** is a network training function that updates weight and bias values per the BFGS quasi-Newton method.

The following figures summarise the differences regard to the performance of the chosen algorithm, it’s training time, and the changes in Mean Squared Error value with respect to the Epoch values.

***Fig.3. Training time for various algorithms***

***Fig.2. Performance over the Epoch values***



***Fig.1.Curve fitting for various algorithms***

***Performance on fitting the output data with targets:***

* Trainlm fits close with a minimum mean squared error with the target attributes for a given set of data. This is a case for all the three different Epoch values.
* With increased Epoch, most of the algorithms are performing better with a sharp decrease in the mean squared error except the gradient descent algorithms (‘traingd’ and ‘traingda’).
* ‘trainlm’

Execution time vs Mean Squared Error for various algorithms