Homework 3: Neural Networks and Regularization

[Points: 12.5 , Issued: 2008/03/20 , Deadline: 2008/05/16 , Tutor: Roland Unterberger ; Infohour: 2008/05/09 , 15:30-16:30 , HS i11 ; Einsichtnahme: 2008/05/30 , 15:30-16:30 , HS i11 ;]

3.1  Simple Regression with Neural Networks (continued)[4 points]

Similar to homework 2.1, a simple 1-dimensional function should be learned with feed-forward neural networks. Use the same data set as for homework 1.

- Train a neural network with \( n = 10 \) neurons. Use the training algorithm 'trainscg', train for 700 epochs. Use the regularized error function \textit{msereg}. Use different regularization factors (\( \alpha \) resp. \textit{net.performParam.ratio} in matlab) of \( \alpha = \{0.9, 0.95, 0.975, 0.99, 0.995, 1.0\} \).
- Plot the mean squared error of the training and of the test set for the given regularization factors.
- Interpret your results. What is the best value of \( \alpha \) ? Also interpret your results with homework 2.1. Is the appropriate selection of the number of hidden neurons, early stopping or the regularized error function the best choice to avoid overfitting in this example? Explain your choice?[1 Extra Point]

3.1.1 Hints

- Normalize your input data using \textit{mapstd}
- This time you CAN NOT use the performance structure returned by the \textit{train} function because it returns the regularized error function and not the mse. Use the \textit{mse} function instead.

3.2  Overfitting with Real-World Data [8.5 points]

In this homework you are supposed to analyse different overfitting avoidance mechanism with neural networks. Use the \texttt{housing.mat} dataset which contains data about the price of houses in boston. The task is to predict the price of unseen houses (regression task). For a detailed description of the dataset see \texttt{housing_description.txt}. You can use the script \texttt{housing_template.m} as template.

- Split the dataset randomly (a useful command is \textit{randperm}) in a training set \( D \) (75%) and a test set \( T \) (25%).
- Train a neural network with \( n = \{2, 5, 10, 20, 40, 60\} \) neurons. Use the training algorithm \textit{trainscg} and learn for 500 epochs

---

1 mailto:neumann@igi.tu-graz.ac.at
2 mailto:caleb@sbox.tugraz.at
3 http://www.igi.tugraz.at/lehre/CI/homework/data/homework1.mat
4 http://www.igi.tugraz.at/lehre/CI/homework/data/housing.mat
5 http://www.igi.tugraz.at/lehre/CI/homework/data/housing_description.txt
6 http://www.igi.tugraz.at/lehre/CI/homework/data/housing_template.m
• Plot the mean squared error of the training and of the test set for the given number of neurons. For the test set, plot the mean squared error (mse) after training ($tmse_{end}$) and the minimum mse during training ($tmse_{min}$). For which number of neurons can we observe (with standard training, i.e. for $tmse_{end}$) underfitting and for which overfitting?

• What is the best number of hidden neurons for standard training ($tmse_{end}$) and what for early stopping ($tmse_{min}$)? Is there a difference between those two numbers? If yes, why?

• Train a neural network with 60 hidden neurons with the regularized error function $msereg$. Use the training algorithm $trainscg$ and learn for 500 epochs. Use the following $\alpha$ values: $\alpha = [0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.75, 1.0]$;

• Plot the mean squared error of the training and of the test set for the given $\alpha$. What is the best $\alpha$? Good working $\alpha$ values for this example are significantly different from example 3.1. Why? For which $\alpha$ can we observe underfitting and for which overfitting?

• Compare the results to standard training and early stopping with varying number of hidden units. Which OF avoidance method would you prefer? Explain your choice!

• Repeat the whole experiments using a 4-fold cross-validation instead of the testset to estimate the true error. Create the same plots and compare them. Which results are more reliable? Does your choice of the optimal number of hidden neurons and the optimal $\alpha$ change?

3.2.1 Hints

– Normalize the data using $mapstd$.

– You can find the source code for splitting the data in a training and a test set in the template.

3.2.2 Remarks

• Present your results clearly, structured and legible. Document them in such a way that anybody can easily reproduce them.

• Please hand in the print out of the Matlab program you have used (no emails!).