Computational Statistics
Neural networks

Download the German Health Care data archive, containing the data file `rwm.data` and the accompanying file

`readme.rwm.txt`.

This dataset is described as follow in Green's book (page 195 of 7th edition):

A recent study in health economics is "Incentive Effects in the Demand for Health Care: A Bivariate Panel Count Data Estimation" by Riphahn, Wambach, and Million (2003). The authors were interested in counts of physician visits and hospital visits and in the impact that the presence of private insurance had on the utilization counts of interest, that is, whether the data contain evidence of moral hazard. The sample used is an unbalanced panel of 7,293 households, the German Socioeconomic Panel (GSOEP) data set.7 Among the variables re- ported in the panel are household income, with numerous other sociodemographic variables such as age, gender, and education. For this example, we will model the distribution of in- come using the last wave of the data set (1988), a cross section with 4,483 observations. Two of the individuals in this sample reported zero income, which is incompatible with the underlying models suggested in the development below. Deleting these two observations leaves a sample of 4,481 observations.

Here, we want to predict the logarithm of income (variable `hhninc`) as a function of age, sex, education and marital status. To apply neural networks, we will use the R package `nnet`.

1. Read the data. Standardize the variables log(income), age and education.

2. Partition the data randomly into a training set and a test set of 1000 observations.

3. Estimate the linear regression analysis on the data, taking log(income) as the dependent variable. Estimate the error on the test data.

4. Using the training set, train different neural networks by varying the number of hidden units and the weight decay parameter λ. Test the performances of the trained neural networks on the test set, and compare them with those of linear regression.

5. Fixing the number of hidden units to 10, optimize λ using 10-fold cross-validation, and compute the test error of the best predictor.