Exercise 1:
Go to `https://www.kaggle.com/t/7f00905fd52d4f4a809d2e342466a054` where you'll find the third deep learning competition for our lecture. This time we have a look at the famous CIFAR-10 data, although we only take at a subset of the data. You'll find a short description of the dataset in the challenge description. If you want to look at the images from the data you can use this code snippet:

```
library(imager)
x = unlist(train[1, -ncol(train)]) / 255
i = as.cimg(array(x, c(32, 32, 1, 3)))
plot(i)
```

Some image preprocessing techniques (rotation, scaling, color changing) might be very useful to increase the amount of training data you have. The `imager` package should be able to do a lot of interesting transformations. Train a deep neural network on the training data `train.csv` and predict `test.csv`. Upload a csv file with the predictions and ids to get your score and place on the leaderboard.

Good Luck!

Note: I know that you could use the full CIFAR-10 dataset for training, but I'd ask you to not do this and rather try to artificially increase the amount of training data using the above mentrioned techniques.

Exercise 2:
We continue the implementation of our home-made neural network. You can either continue with your own implementation or use my solution from the last exercise sheet. Either way extend the code to do the following things:

- Add support of multi class classification
- Allow L2-Regularization with a λ parameter to control the amount of regularization
- Implement the momentum method for accelerated gradient descent learning.

Exercise 3:
How do the eigenvalues of the Hessian matrix of the weights and the λ parameter of L2-regularization play together to shrink the weights in each gradient descent step. Derive a a formula showing the rescaling of the weights and intrepret the formula.

Hint: Start with a quadratic Taylor-approximation at the minimum of the cost function.