# Week 9:

## Que Tran

## 25th April 2017

| Time | Topic | Reading |
|------|-------|---------|
| 8.15- | Recap of tutorials last week | |
| | Lecture: Overfitting and Underfitting | |
| 9.00- | Exercises on Overfitting and Underfitting | |
| 11.45- | Lunch break | |
| 12.15- | Questions and answers | |
| | | |
| 13.00- | Continue with previous exercises | |

Rest of Day Catch up with previous tutorials

This PDF document is available in an HTML version at `http://www.hg.schaathun.net/FPIA/week09.html`.

# 1 Early stopping

At this point, the priority is to complete exercises implementing and using the neural network.

If you have time, the following exercises can give practice on the techniques from today's lecture.

In this exercise, we will implement *Early stopping* method to avoid overfitting.

## 1.1 Split the data

In this technique, you need to divide the available data three subsets: training, validation and test set. Write a function that randomly splits your data into 3 subsets.

If you need a hint, come back to **Tutorial 4.1** and take a look at `mkTestTrainSets`. What you need to do is pretty much similar.

## 1.2 Monitor the error on the validation set

Modify function `trainNetwork` to monitor the error on the validation set during training process. The process should stop when the validation error increases.

- First, try to stop the training process right after validation error starts increasing.

- Now modify your `trainNetwork` a bit so that the training process will be stopped when the validation error increases for a specified number of iterations, called `maxFail`. When the training process is stopped, the network at minimum of validation error is returned. You can set `maxFail = 6` as a default value and try out different values of `maxFail` afterwards.

- Test your networks with the test set and compare the results from the two experiments. Which one gives you better results? Can you explain the reason?