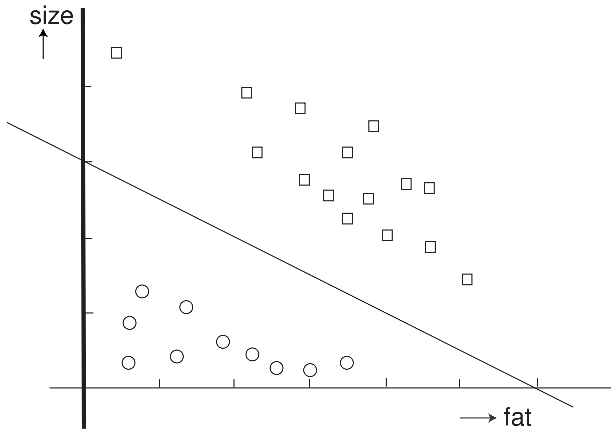


Figure 1: The health expert's decsion boundary

## Exercise sheet 1: Applying the perceptron algorithm (1/3)

Suppose we work in a chips factory. Two important aspects for quality control are size of the individual chips and their fat content. Due to increasing worries about obesity only small chips are allowed to have a large fat percentage. If the chip is large it must have a lower fat content. A health expert works in the factory to ensure that food standards are enforced. In figure 1 we see the fat percentage and the size of individual chips, which have been marked by the health expert as 'healthy' and as 'unhealthy'. With a ruler the expert has drawn a straight line that separates the two sets. The chip manufacterer wants to dispense with the health expert because of financial concerns. Your task is to develop a neural network that can replace the expert and automatically determines whether a chip is 'healthy' or 'unhealthy'. You go to work and based on the health expert's linear relationship, you develop a perceptron which automates the decision process.

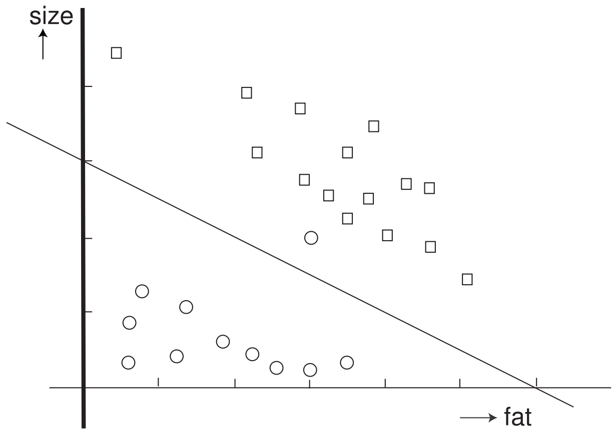


Figure 2: One of the chips is misclassified!

### Exercise sheet 1: Applying the perceptron algorithm (2/3)

*Do this. Find weights and bias for a perceptron based on the linear relation in Figure 1*

It turns out that external quality control reveals that sometimes our classifier doesn't work correctly! In figure 2 we see a point that is misclassified by the straight line. In order to correct automatically, we invoke the perceptron algorithm.
*Use the perceptron algorithm to adapt weights and threshold of the perceptron. Carry out a single step of the algorithm. Start with the weights found above. Assume that the misclassified point is at (3,2) exactly. Is the solution satisfactory? Are more steps needed?*
If you have carried out the step above, you will find that the new solution is not very good. The resulting line doesn't separate the two sets. The lesson to be learned from this is that the perceptron algorithm has to be applied over and over again *on all points* until *all points are classified correctly.* This is somewhat unsatisfactory. From a situation where we nearly had a solution, we to start all over again. In the optional part on learning rate below, we will see that it is in fact possible to be a bit more subtle. First consider the following question, however:

## Exercise sheet 1: Applying the perceptron algorithm (3/3)

*Does it make a difference for the function of the MP neuron if we multiply all weights and the bias by the same constant? Why?*
Suppose that intensive research in health care has resulted in the fact that the sets of healthy and unhealthy chips is linearly separable.
*What does this mean ?*
*Outline a procedure to dispense with expert for good, i.e. explain how the perceptron algorithm can be used to correct flukes automatically (why is the alinea above important? )*

**Learning rate (optional)**

If you have considered the question of whether multiplying the weights and the threshold by the same fixed constant makes a difference, you should have concluded that the answer is 'no'. After all, such parameters describe the same straight line! Now imagine the situation where we multiplied the weights and threshold by a factor of 100. Now, a single step of the perceptron algorithm has the same influence on the line parameters in absolute terms, but because the line parameters are much bigger, the relative influence of a single step of the algorithm is much smaller.
Argue that the same effect might have been reached if we change the step:

$$w \to w \pm x \tag{1}$$

to

$$w \to w \pm \lambda x, \tag{2}$$

where $\lambda$ is an arbitrary constant. It is called the learning rate and can be chosen to be small. In this way a single learning step doesn't affect the weights and the bias so much.
*Try a learning rate of 0.1*