Leren homework # 4

Date: December 1, 2014

Name (Student Number):
M. Pfundstein (10452397)
T.M. Meijers (10647023)

Question 1

(a) $\theta : \{\mu, \sigma^2\}$

Bias: $b_\theta(d) = E[d] - \theta$

Variance: $E[(d - E[d])^2]$

Where d is the estimator of θ and E the expectation. The bias is thus defined as the difference of what our parameters actually predict and the *expected* value that our parameters *should* predict (In real world examples we don't know this though). A large bias indicates therefore that we are quite off from the real world. The variance is defined as the expected squared difference between the estimator and the expected estimator. A larger variance therefore indicates that we *expect* much room between what we will estimate and what we *expect* to estimate.

(b) Yes, with the following formulas:

$\overline{h}(x) = \frac{1}{M} \sum_{t=1}^{M} h_i(x)$

$Bias^2(h) = \frac{1}{N} \sum_{t=1}^{N} [\overline{h}(x^t) - f(x^t)]^2$

$Variance(g) = \frac{1}{NM} \sum_{t=1}^{N} \sum_{i=1}^{M} [h_i(x^t) - \overline{h}(x^t)]$ (g?)

With a neural network and a dataset we have access to all the variables needed to fill in the formulas above.

(c) If we increase the number of hidden nodes (complexity), we observe that the bias will decrease (we can fit the data better). However, variance will increase (since our hyptohesis will become a higher-order polynomial and thus fit the data more accurately). Hence it is a tradeoff between under- and overfitting.

(d) Yes. We can increase the sample size (decrease variance) *and* add a hidden node to your neural network (decrease bias). But adding the hidden node would also increase our variance again. Thus this process must be done very carefully and in such a way that the decrase in variance is 'larger' than the increase. In practice this is mostly not possible (Bias-variance dilemma). It probably requires precise tuning with the regularization parameter etc.
Example: Consider a sample of features that has a very parabolic-like layout. Now if we fit a linear function to the data, we get a very high bias and also much variance. If we now add a quadratic term and the trained hypothesis fits *perfectly* the dataset, we would get an immediate decrease in variance *and* bias.

Question 2

(a) See Figures 2, 3, 4 and 1 together with 5 for the corresponding tree.

(b) When examining the decision boundaries of the function, we cannot really say which one is the *best* one. There are only eleven data points and there is also no validation/test set available. If we approach the problem logically then the candidates for the best algorithm are the quadratic logistic regression and 1-nearest neighbour with a small favour for 1-nearest neighbour. Both algorithm show that in the upper middle to the upper right of plot, the positive instances can be found. The remainder of the plot are negative instances. Quadratic logistic regression does incorrectly label 1 positive instance as a negative instance and 1 negative instance as a postive instance. This does not happen with 1-nearest neighbour. Still we can't be sure if the other points in this area are positive or negatives (we need a bigger dataset), so 1-nearest neighbour could be wrong there.

The reason why we not include decision tree in our set of the *best* ones is that the decision boundaries are too straight. When we look at the quadratic logistic regression boundary and combine it with the boundary from the 1-nearest neighbour, then we see that the area of positive examples gets smaller the more we approach the right side. The decision tree boundary doesn't take that into account. Hence it feels to artifically.
Linear logistic regression gets ruled out because of its obviously high bias and the non linearity of the dataset.

Question 3

(a) No answer required. (No question asked)

(b) See Figure 6 for the tree itself.

Entropy before adding A1:

$$E(S) = -(2 \cdot \frac{3}{6} \cdot log_2(\frac{3}{6})) = 1$$

After adding A1:

$$\begin{aligned} E(X < 8.5) &= -\frac{1}{3}log(\frac{1}{3}) - \frac{2}{3}log(\frac{2}{3}) = 0.918 \\ E(X \geq 8.5) &= -\frac{2}{3}log(\frac{2}{3}) - \frac{1}{3}log(\frac{1}{3}) = 0.918 \\ Gain &= E(S) - (\frac{3}{6}E(X < 8.5) + \frac{3}{6}E(X \geq 8.5)) \\ &= 1 - 0.918 = 0.082 \end{aligned}$$

Cost:

$$\begin{aligned} \theta_0 &= -8.5 \\ \theta_1 &= 1 \end{aligned}$$

$$
\begin{aligned}
J_{8.5} = \frac{1}{6}(&log(1-h(-8.5+1)) \\
&+log(1-h(-8.5+3))) \\
&+log(h(-8.5+7)) \\
&+log(h(-8.5+10)) \\
&+log(1-h(-8.5+11)) \\
&+log(h(-8.5+15))) \\
= \frac{1}{6}(&-0.0006 \\
&-0.0041 \\
&-1.7014 \\
&-0.2014 \\
&-2.5789 \\
&-0.0015) = 0.7480
\end{aligned}
$$

Accuracy: 4 out of 6 are correct after adding A1, so accuracy $= 4/6 \approx 66\%$

(c) If we are setting a single class boundary (So only adding A1, not other nodes) there would be better class boundaries. For example if one takes a class boundary of 6:

$E(S) = -\frac{2}{6} \cdot (-1 \cdot log_2(1)) - \frac{4}{6} \cdot (\frac{3}{4} \cdot log_2(\frac{3}{4}) + \frac{1}{4} \cdot log_2(\frac{1}{4})) \approx 0.541$
This is way better than a boundary of 8.5 (Entropy of 0.541 versus 0.918), while accuracy is also higher with 5/6 correct as opposed to 4/6.

.

Figures

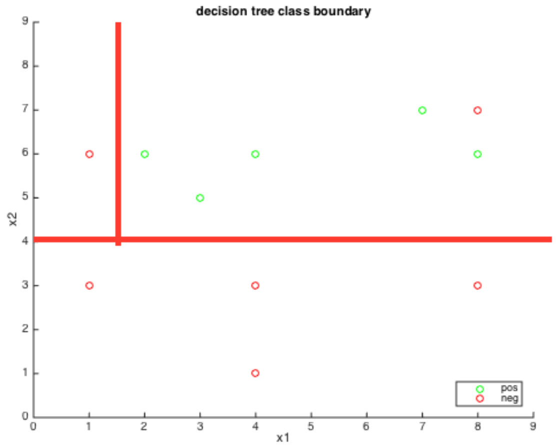

Figure 1: Decision Tree

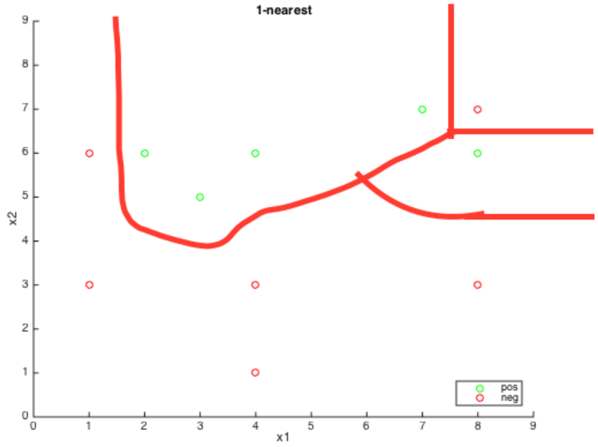

Figure 2: 1-nearest neighbour

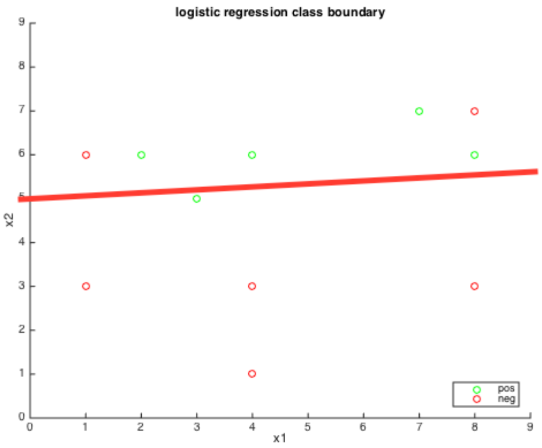

Figure 3: Logistic Regression

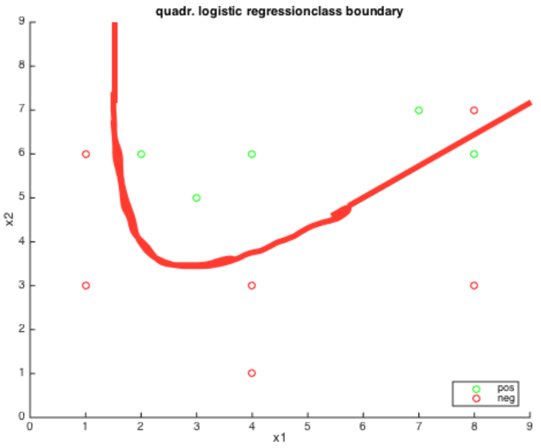

Figure 4: Logistic Regression with Quadratic Terms

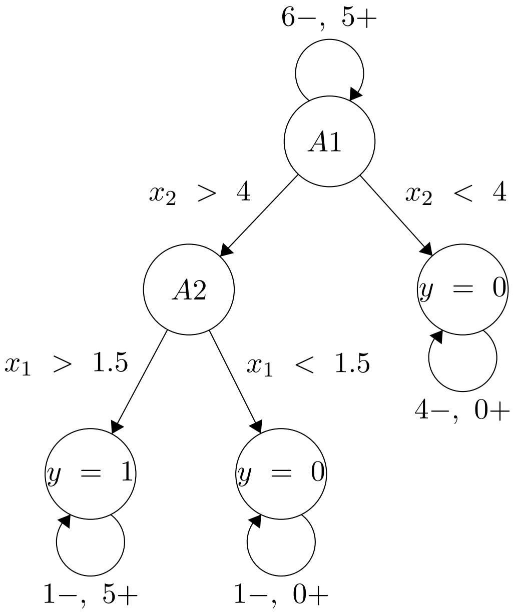

Figure 5: Q2.a decision tree

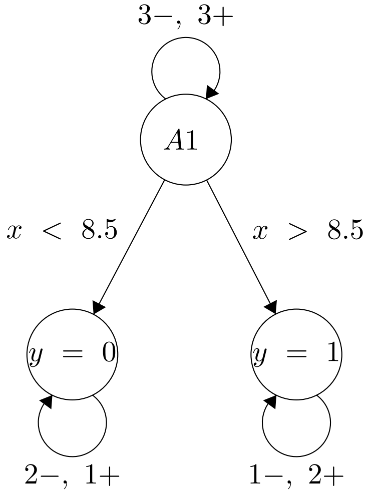

Figure 6: Q3.b decision tree