Linear models

**Names including ID:  
 1:Martin van Leeuwen 0901497  
 2:Frouke Hekker 0897373**

Q1 :

Q2 :

When is redefined as [w;b] and x=[x,1], where this 1 represents a row of ones as large as x, then the formula becomes .Since matrix multiplications work by multiplying the elements of the first matrix’s row with the seconds column it can be said that for each element in the new matrix the formula will be: where n and m are the size of . Doing this for each element corresponds to .

This means that now again the formula needs to be minimalized in order to get the lowest possible cost. Just like in the previous exercises and lecture 1 this corresponds to the Wiener filter:

Q3:

Implementing the previous answer on this data yields . The loss of the model is equal to . Since the loss is very low we can conclude that the process is well described by the model.

Q4: With the new y the optimal parameters yield . As a consequence of the noise the loss has increased to 0.27 .Regularization techniques can be applied in order to get a model that is more robust to noise.

regularization has a smoothing effect on the weights making it less vulnerable to noise. The learning algorithm perceives higher variance of X , causing it to shrink weights of features that have low covariance compared to the added variance.

Furthermore, by using more datapoints to train the filter the resulting model may also become a better description of the input process.

Q5:

The derivative of the cost function will have a different size in the dimension which corresponds to the covariance. These dimensions, or weights, will converge faster is the derivative is higher. The cost function will converge faster is the covariance’s are all the same, since we can move in a straight line to the minimum.

Q6:

When the regression model is trained on the XOR function the optimal parameters become

the resulting loss is 2.0.

Nonlinear functions

Q7:

ReLu :

Sigmoid:

Softmax:

Q8:

ReLu: for all x large than zero the derivative is 1.

Sigmoid: if x gets large the derivative approaches zero

Softmax: This function is not dependent on the values of x but only on the ratios between the elements of x. Thus it will give a good gradient regardless of x values.

Shallow nonlinear models

Q9: The outputs of the XOR cannot be classified effectively using a linear decision boundary. Even if multiple linear transformations are combined the resulting decision boundary will remain linear. However, by applying a non-linear transform the inputs are in essence mapped to a non-linear coordinate system. By picking the right transform we can make it easier to separate the input points using a linear decision boundary.

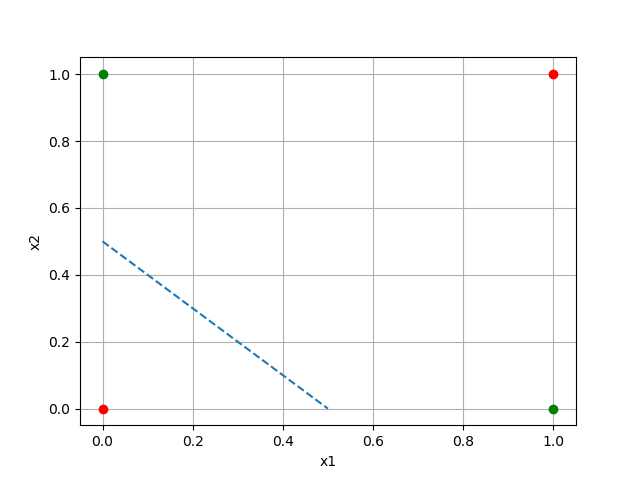
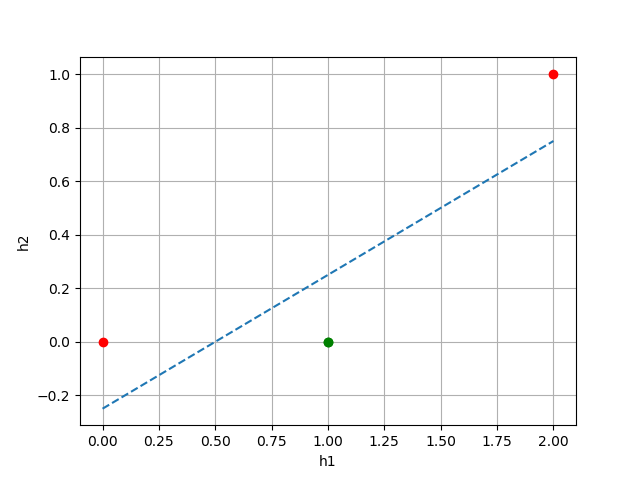


Figure 2, latent space XOR function, red = false,

green = true

Figure 1, input space of XOR function, red = false,green = true

Q10, Q11: by applying function 1.3 the input space depicted in figure 1 is transformed to the latent space in figure 2.

Binary classification with logistic regression

Q12: A softmax function gives the probability of the input belonging to each class. Consequently, the input can be classified by picking the class with the highest probability. That is why a softmax function is useful for categorical classification.

Q13:

Q14:

Q15:

This step relates to the update of the weight w. The final loss gradient that is derived shows how much each weight needs to update in order to get a lower loss the next time the network is used

Q16:

Classification with a shallow nonlinear model

Q17: