Exercise 10 – Neural Networks

Introduction to Machine Learning

Exercise 1: Logistic regression or deep learning?

Learning goals

1) Understand how stacking single neurons can solve a non-linear classification problem
2) Translate between functional & graph description of NNs

Suppose you have a set of five training points from two classes. Consider a logistic regression model $f(\mathbf{x}) = \sigma\left(\alpha^\top \mathbf{x}\right) = \sigma(\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2)$, with $\sigma(\cdot)$ the logistic/sigmoid function, $\sigma(c) = \frac{1}{1+\exp(-c)}$.

Which values for $\alpha = (\alpha_0, \alpha_1, \alpha_2)^\top$ would result in correct classification for the problem in Figure 1 (assuming a threshold of 0.5 for the positive class)?

Don't use any statistical estimation to answer this question – think in geometrical terms: you need a linear hyperplane that represents a suitable decision boundary.

Apply the same principle to find the parameters $\beta = (\beta_0, \beta_1, \beta_2)^\top$ for the modified problem in Figure 2.

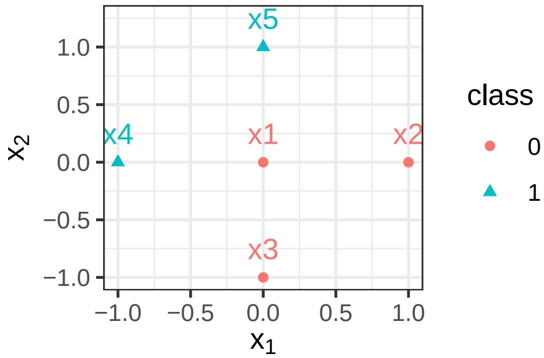

Figure 1: Classification problem I

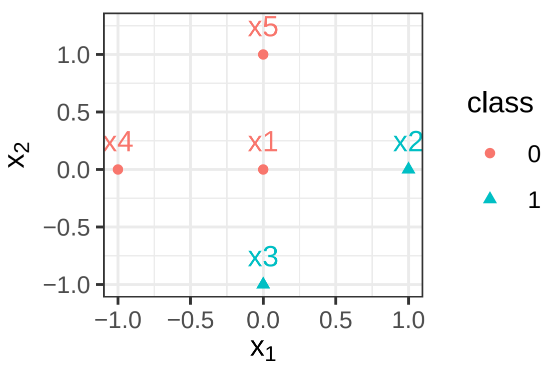

Figure 2: Classification problem II

Now consider the problem in Figure 3, which is not linearly separable anymore, so logistic regression will not help us any longer. Suppose we had alternative coordinates $(z_1, z_2)^\top$ for our data points:

i	$z_1^{(i)}$	$z_2^{(i)}$	$y^{(i)}$
1	0	0	1
2	0	1	0
3	0	1	0
4	1	0	0
5	1	0	0

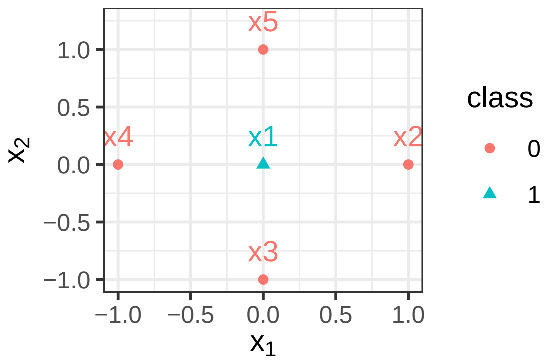

Figure 3: Classification problem III

Explain how we can use z_1 and z_2 to classify the dataset in Figure 3. The question is now, of course, how we can get these z_1 and z_2 that provide a solution to our previously unsolved problem – naturally, from the data.

The question is now, of course, how we can get these z1 and z2 that provide a solution to our previously unsolved problem – naturally, from the data.

Perform logistic regression to predict z_1 and z_2 (separately), treating them as target labels to the original observations with coordinates $(x_1, x_2)^\top$. Find the respective parameter vectors $\gamma, \phi \in \mathbb{R}^3$.

Lastly, put together your previous results to formulate a model that predicts the original target y from the original features $(x_1, x_2)^\top$.

Sketch the neural network you just created (perhaps without realizing it).

Explain briefly how the chain rule is applied to the computational graph such a neural network represents. Can you think of a use we can put the resulting gradients to?