Connectionist Neurons and Multi Layer Perceptrons

Exercise T2.1: Terminology (tutorial)

(a) How does a nonlinear transfer function change the computational properties of a connectionist neuron? In which situations might this be useful?

(b) Which effect has the *bias* in a connectionist neuron? Give an example in which a classification with a `sign`$(\cdot)$ transfer function would *not* work without a bias (but would with one).

(c) What are *point* and *edge filters* and what are they used for?

(d) What is the difference between a *connectionist neuron* with a *logistic transfer function* and a *stochastic neuron*?

(e) What are *feedforward* and *recurrent* MLP and why is training a recurrent MLP much more complicated than a feedforward MLP?

Exercise H2.1: Connectionist Neuron (homework, 6 points)

The dataset `applesOranges.csv` available on ISIS contains 200 measurements (`x.1` and `x.2`) from two types of objects as indicated by the column `y`. In this exercise, you should use a simple connectionist neuron with the sign function as transfer function to classify the objects i.e.

$$f(\mathbf{x}) = \mathrm{sgn}(\underline{\mathbf{w}}^T \underline{\mathbf{x}} - \theta)$$

(a) Plot the data in a scatter plot (x.1 vs. x.2). Use color to indicate the type of each object.

(b) Set $\theta = 0$. Create a set of 19 equally spaced weight vectors $\underline{\mathbf{w}} = [w_1, w_2]$ on the circle centered on $(0, 0)$ with radius 1. I.e. if α denotes the angle between the weight vector and the x-asis, for each weight $||\underline{\mathbf{w}}|| = 1$ and $\alpha_1 = 0, \alpha_2 = 10, ..., \alpha_{19} = 180$ such that $w_1 \in [-1, 1], w_2 \in [0, 1]$. For each weight vector $\mathbf{w}$ determine the classification performance ρ (% correct classifications) of the corresponding neuron and plot a curve showing α vs. ρ.

(c) From these weights, pick the weight vector yielding best performance. Now vary $\theta \in [-3, 3]$ and pick the value of θ giving the best performance.

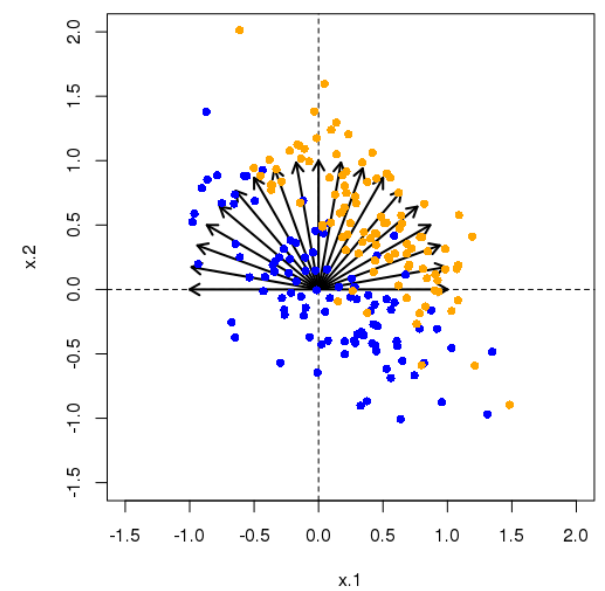
(d) Plot the datapoints, colored according to the classification corresponding to these parameter values. Plot the weight vector $\underline{\mathbf{w}}$ in the same plot. How do you interpret your results?

(e) Find the best combination of $\underline{\mathbf{w}}$ and θ by exploring all combinations of α and θ (within a sensible range and with sensible precision) and plotting the performance of all combinations in a heatmap.

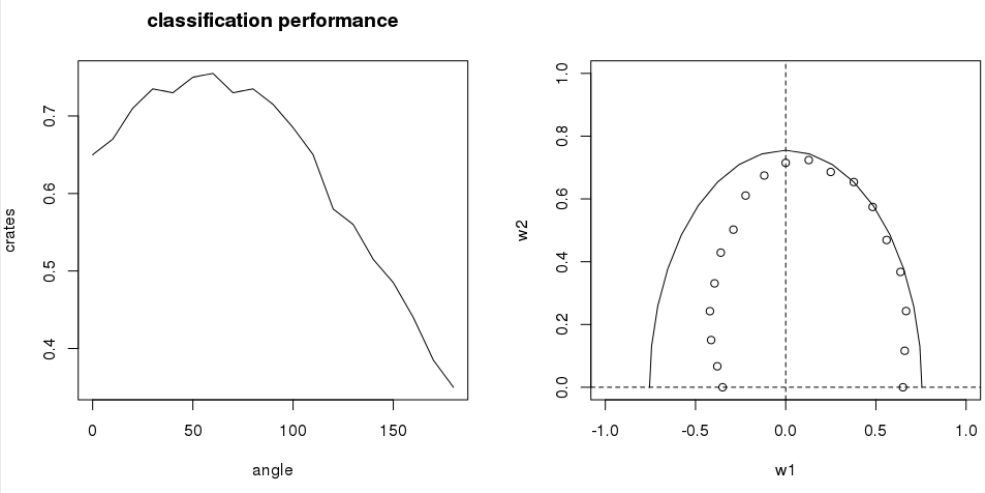
(f) Can the optimization method (e) be applied to any classification problem? Discuss potential problems and give an application example in which the above method must fail.

Solution

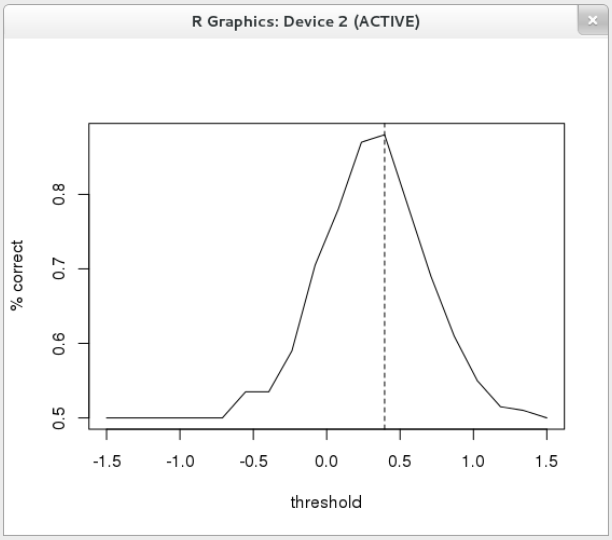
(a) The labelled data and the 19 weight vectors from (b).



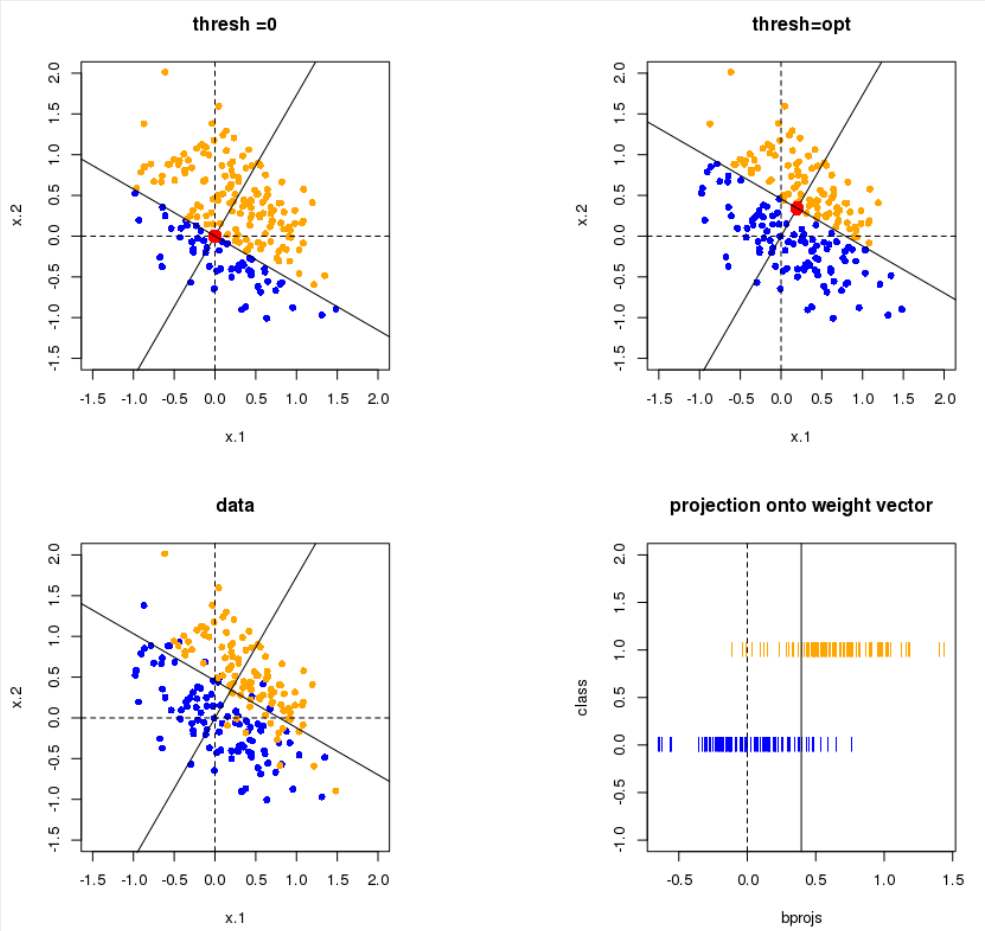
(b) The left plot shows the classification performance vs. the angle α, and the right plot shows that performance in a radial plot (circles).



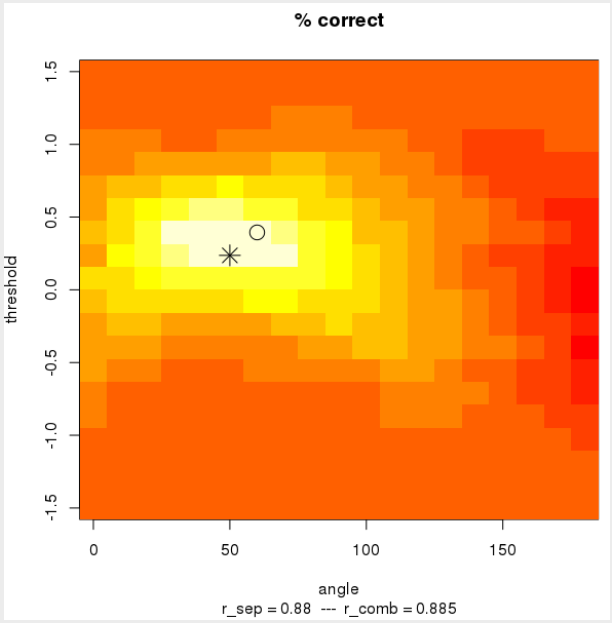
(c) The performance at the best angle from (b) for varying thresholds θ.



(d) The decision boundaries for thresholds $\theta = 0$ (upper left plot), and the optimal θ from (c) (upper right plot). The latter is plotted against the original labels in the lower left. The lower right plots the projection of the data points onto the weight vector.



(e) The heat-map of correct classification for range of angles (x-axis) and thresholds (y-axis).



(f) Grid-search fails in high dimensional input spaces, as the number of measured combinations grows exponential in the inputs dimensionality.

Exercise H2.2: Multi-layer Perceptrons (homework, 4 points)

(a) Describe a simple example in which a *multilayer perceptron* (MLP) can distinguish between two classes, but a single connectionist neuron can not.

(b) For a MLP with input $x \in \mathbb{R}$ and one hidden layer, the input-output function can be computed as

$$y(x) = \sum_{i=1}^{n_{\text{hid}}} w_i f(a_i(x - b_i))$$

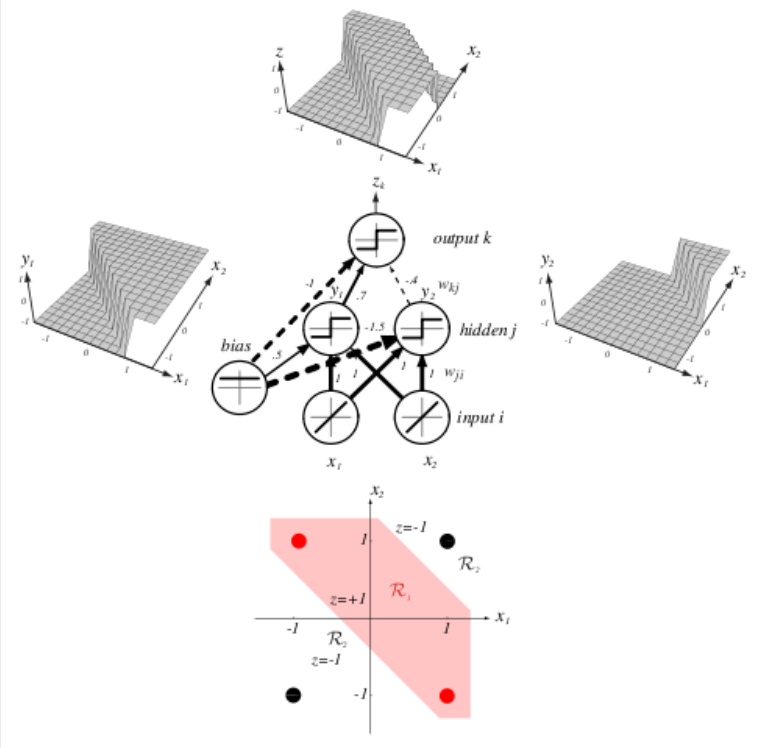
with output weights w_i and parameters a_i and b_i for each hidden unit i. Create 50 MLPs with $n_{\text{hid}} = 10$ hidden units by sampling for each one a set of random parameters $\{w_i, a_i, b_i\}, i = 1, ..., 10$ and using $f := \tanh$ as the activation function. Use $a_i \sim \mathcal{N}(0, 2), w_i \sim \mathcal{N}(0, 1)$ and uniformly distributed $b_i \sim \mathcal{U}(-2, 2)$. Plot the input-output functions of these 50 MLPs for $x \in [-2, 2]$.

(c) Repeat this procedure using instead $a_i \sim \mathcal{N}(0, 0.5)$. What is the difference?

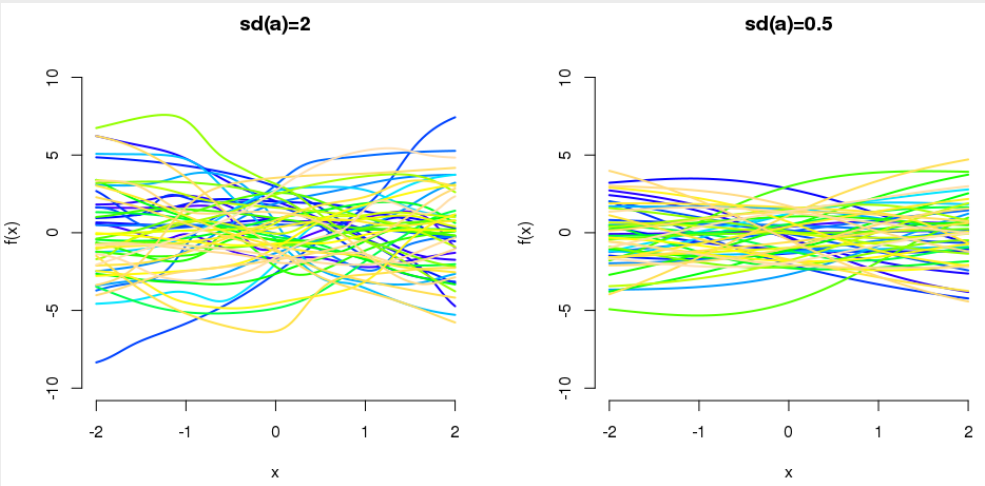
(d) Compute the mean squared error between these 2x50 input-output functions and the function $g(x) = -x$. Which MLPs from these two classes approximate it best? Plot these 2 functions.

Solution

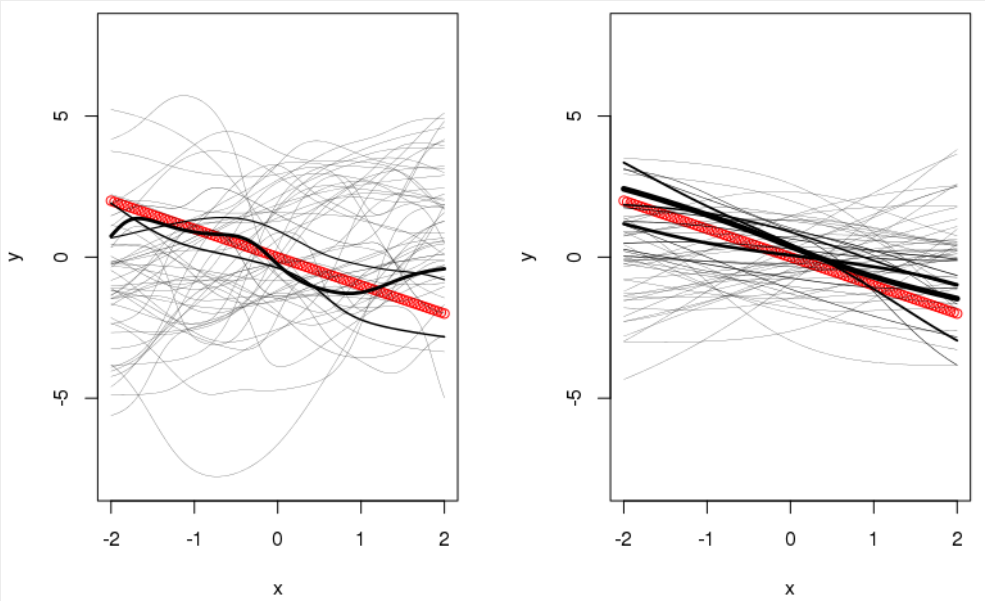
(a) The logical extended-or (XOR) is a classical example that is not linearly separable (from Duda et al., 2000).



(b) is plotted on the left side and (c) on the right side. The difference is the smoothness of the drawn functions.



(d) The MLPs from (b) and (c) are plotted with a thickness corresponding inversely to their me mean squared error for training data from function g, plotted in red. The thickest two lines are the two MLP that approximate g the best.



Total 10 points.