Machine Intelligence 1
WS 2017/18, Obermayer/Augustin

Exercise Sheet 2
due: 01.11.2017 at 23:55

Connectionist Neurons and Multi Layer Perceptrons

Please remember to upload exactly *one* ZIP file per group and name the file according to the respective group name: `yourgroupname.zip`
The ZIP file should contain a single Jupyter notebook source file as well as a single PDF file that is generated from the Jupyter notebook. Please no folder structure, exercise PDF or data files.

Exercise T2.1: Terminology (tutorial)

(a) What does a connectionist neuron compute?

(b) Which effect have the *weights* and the *bias*, respectively?

(c) Why is a nonlinear transfer function beneficial compared to a linear one?

(d) What is a feedforward multilayer perceptron (MLP)?

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

The dataset[1] `applesOranges.csv` contains 200 measurements (`x.1` and `x.2`) from two types of objects as indicated by the column `y`. In this exercise, you shall use a simple connectionist neuron with the sign function as transfer function to classify the objects, i.e., obtain the predicted class $\hat{y}$ for a data point $\underline{\mathbf{x}} \in \mathbb{R}^2$ by

$$\hat{y}(\underline{\mathbf{x}}) = \widetilde{\operatorname{sgn}}(\underline{\mathbf{w}}^T \underline{\mathbf{x}} - \theta)$$

with slightly modified sign function

$$\widetilde{\operatorname{sgn}}(h) = \begin{cases} 1 \text{ for } h \geq 0, \\ 0 \text{ otherwise.} \end{cases}$$

(instead of the usual +1/-1 sgn function), where $h = \underline{\mathbf{w}}^T \underline{\mathbf{x}} - \theta$ is the input to the neuron.

(a) Plot the data in a scatter plot (x_1 vs. x_2). Use color to indicate the type of each object.

(b) First, set the bias $\theta = 0$. Create a set of 19 weight vectors $\underline{\mathbf{w}} = (w_1, w_2)^T$ pointing from the origin to the upper semi-circle with radius 1. I.e. if α denotes the angle between the weight vector and the x-axis, for each $\alpha = 0, 10, \ldots, 180$ (equally spaced) such that $||\underline{\mathbf{w}}||_2 = 1, w_1 \in [-1, 1], w_2 \in [0, 1]$. For each of these weight vectors $\underline{\mathbf{w}}$ determine the classification performance ρ (% correct classifications) of the corresponding neuron and plot a curve showing ρ as a function of α.

(c) From these weight vectors, pick the $\underline{\mathbf{w}}$ yielding best performance. Now vary the $\theta \in [-3, 3]$ and pick the value of θ giving the best performance.

(d) Plot the data points, 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?

[1]This data file (and those required for future exercise sheets) is available on ISIS.

(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.

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

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

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

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 transfer function. Use normally distributed $a_i \sim \mathcal{N}(0, 2), w_i \sim \mathcal{N}(0, 1)$ and uniformly distributed $b_i \sim \mathcal{U}(-2, 2)$. Plot the input-output functions of these 50 MLPs for $x \in [-2, 2]$.

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

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

Total 10 points.