Machine Intelligence 1 | **Exercise Sheet 2**
WS 2022/23, Obermayer/Kashef/Strömsdörfer | due: 10.11.2022 at 23:55

Connectionist Neurons & Function Fitting

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 do **not** include any folder structure, exercise PDF or data files.

Exercise T2.1: Function Fitting (tutorial)

(a) What effect will the choice of error measure (particularly *quadratic or linear*) produce?

(b) Outline the relation between the quadratic error function and the Gaussian conditional distribution for the labels.

(c) Derive a suitable error function (*cross entropy*) for the following case: the output of a neural network is interpreted as the probability that the input belongs to the first of two classes.

Exercise T2.2: Gradient Descent (tutorial)

(a) Outline gradient descent.

(b) Determine the update rule for gradient descent for a connectionist neuron.

Exercise H2.1: Connectionist Neurons (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 will use a connectionist neuron with a "binary" transfer function $f(h)$ to classify the objects, i.e., obtain the predicted class y for a data point $\underline{\mathbf{x}} \in \mathbb{R}^2$ by

$$y(\underline{\mathbf{x}}) := f(\underline{\mathbf{w}}^\top \underline{\mathbf{x}} - \theta)$$

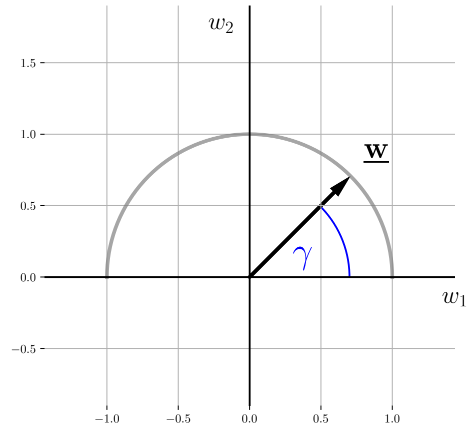
with

$$f(h) := \begin{cases} 1 & \text{for } h \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

where $h := \underline{\mathbf{w}}^\top \underline{\mathbf{x}} - \theta$ is the total input to the neuron.

(a) Plot the data in a scatter plot (x_2 vs. x_1). Mark the points with different colors to indicate the type of each object.

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



(b) Set the bias $\theta = 0$. Create a set of 19 weight vectors $\underline{\mathbf{w}} = (w_1, w_2)^\top$ pointing from the origin to the upper semi-circle with radius 1 (i.e. if γ denotes the angle between the weight vector and the x-axis, for each $\gamma = 0, 10, \dots, 180$ (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 $\underline{\mathbf{w}}$,

  (i) determine $\%$ correct classifications ρ of the corresponding neuron and

  (ii) plot a curve showing ρ as a function of γ.

(c) Out of the 19 weight vectors from above, pick the $\underline{\mathbf{w}}$ that yields the best performance. Now, vary the bias $\theta \in [-3, 3]$ and pick the value of θ that gives the best performance.

(d) Plot the data points and color them according to the predicted classification when using the $\underline{\mathbf{w}}$ and θ that led to the highest performance. 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 γ and θ (within a reasonable range and precision). Compute and plot the performance of all combinations in a heatmap.

(f) Can the *grid-search* optimization procedure used in (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: Binary Classification (homework, 4 points)

For binary targets $y_T^{(\alpha)} \in \{0, 1\}$ the network output $y(\underline{\mathbf{x}}; \underline{\mathbf{w}}) \in (0, 1)$ can be interpreted as a probability $P(y = 1|\underline{\mathbf{x}}; \underline{\mathbf{w}})$. A suitable error function for this problem is:

$$E^T = \frac{1}{p} \sum_{\alpha=1}^{p} e^{(\alpha)}$$

with

$$e^{(\alpha)} = -\left[y_T^{(\alpha)} \ln y(\underline{\mathbf{x}}^{(\alpha)}; \underline{\mathbf{w}}) + (1 - y_T^{(\alpha)}) \ln \left(1 - y(\underline{\mathbf{x}}^{(\alpha)}; \underline{\mathbf{w}}) \right) \right].$$

(a) (1 point) Show that

$$\frac{\partial e^{(\alpha)}}{\partial y(\underline{\mathbf{x}}^{(\alpha)}; \underline{\mathbf{w}})} = \frac{y(\underline{\mathbf{x}}^{(\alpha)}; \underline{\mathbf{w}}) - y_T^{(\alpha)}}{y(\underline{\mathbf{x}}^{(\alpha)}; \underline{\mathbf{w}}) \left(1 - y(\underline{\mathbf{x}}^{(\alpha)}; \underline{\mathbf{w}}) \right)}$$

(b) (1 point) Consider an MLP with one hidden layer. The nonlinear transfer function for the output neuron ($i = 1, v = 2$) is assumed to be

$$f(h_1^2) = \frac{1}{1 + \exp(-h_1^2)},$$

where h_1^2 is the total input[2] of the output neuron. Show that its derivative can be expressed as

$$f'(h_1^2) = f(h_1^2) \left(1 - f(h_1^2) \right).$$

(c) (1 point) Using the results from (a) and (b), show that the gradient of the error function $e^{(\alpha)}$ with respect to the weight w_{1j}^{21} between the the single output neuron ($i = 1, v = 2$) and neuron j of the hidden layer ($j > 0, v = 1$) is

$$\frac{\partial e^{(\alpha)}}{\partial w_{1j}^{21}} = \left(y(\underline{\mathbf{x}}^{(\alpha)}; \underline{\mathbf{w}}) - y_T^{(\alpha)} \right) f(h_j^1).$$

Total 10 points.

[2] The total input of a neuron is sometimes referred to as a *logit*.