Machine Learning Group
Faculty IV – Electrical Engineering and Computer Science
Technische Universität Berlin

Exercise Sheet 9

Exercise 1: Computing Gradients in RNNs (5×10 + 5×10 = 100 P)

We consider the task of binary classifying univariate time series (only two time steps for the purpose of the exercise) using a recurrent neural network. Let (x_1, x_2) be the time series given as input. The recurrent neural network is given by the equations:

$$\begin{aligned} h_1 &= w \cdot x_1 + \tanh(h_0) \\ h_2 &= w \cdot x_2 + \tanh(h_1) \\ y &= h_1 + h_2, \end{aligned}$$

and we assume that the neural network has initial state $h_0 = 0$. The variable y is the neural network output and w is the model parameter. We further assume that the univariate time series (x_1, x_2) comes with a binary target label $t \in \{-1, 1\}$ and the prediction error for this data point is modeled via the log-loss function

$$\mathcal{L}(y, t) = \log(1 + \exp(-yt)).$$

We would like to extract the gradient of the objective w.r.t. the parameter w.

(a) Draw the neural network graph, and annotate it with relevant variables (inputs, activations, and parameters).

(b) *Compute* $\partial \mathcal{L} / \partial y$.

(c) Assuming the last computation was stored in g, *compute* $\partial \mathcal{L} / \partial h_2$ as a function of g.

(d) Assuming the last computation was stored in δ_2, *compute* $\partial \mathcal{L} / \partial h_1$ as a function of g and δ_2.

(e) Assuming the last computation was stored in δ_1, *compute* $\partial \mathcal{L} / \partial w$ as a function of g, δ_2 and δ_1.

(f) Repeat the steps above (a–e) for the case where the recurrent neural network is given by the equations:

$$\begin{aligned} h_1 &= \tanh(x_1 + w + h_0) \\ h_2 &= \tanh(x_2 + w + h_1) \\ y &= h_1 + h_2, \end{aligned}$$

where the initial state is set to $h_0 = 0$, the target is real-valued ($t \in \mathbb{R}$), and the error function is given by

$$\mathcal{L}(y, t) = \log\cosh(y - t).$$