Exercises for the course
Machine Learning 1
Winter semester 2023/24

Fachgebiet Maschinelles Lernen
Institut für Softwaretechnik und theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

Exercise Sheet 9

Exercise 1: Neural Network Optimization (15 + 15 P)

Consider the one-layer neural network

$$y = \boldsymbol{w}^\top \boldsymbol{x} + b$$

applied to data points $\boldsymbol{x} \in \mathbb{R}^d$, and where $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are the parameters of the model. We consider the optimization of the objective:

$$J(\boldsymbol{w}) = \mathbb{E}_{\hat{p}}\Big[\frac{1}{2}(1 - y \cdot t)^2\Big],$$

where the expectation is computed over an empirical approximation $\hat{p}$ of the true joint distribution $p(\boldsymbol{x}, t)$ and $t \in \{-1, 1\}$. The input data follows the distribution $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I)$ where $\boldsymbol{\mu}$ and σ^2 are the mean and variance.

(a) *Compute* the Hessian of the objective function J at the current location $\boldsymbol{w}$ in the parameter space, and as a function of the parameters $\boldsymbol{\mu}$ and σ of the data.

(b) *Show* that the condition number of the Hessian is given by: $\dfrac{\lambda_1}{\lambda_d} = 1 + \dfrac{\|\boldsymbol{\mu}\|^2}{\sigma^2}$.

Exercise 2: Neural Network Regularization (10 + 10 + 10 P)

For a neural network to generalize from limited data, it is desirable to make it sufficiently invariant to small local variations. This can be done by limiting the gradient norm $\|\partial f / \partial \boldsymbol{x}\|$ for all $\boldsymbol{x}$ in the input domain. As the input domain can be high-dimensional, it is impractical to minimize the gradient norm directly. Instead, we can minimize an upper-bound of it that depends only on the model parameters.

We consider a two-layer neural network with d input neurons, h hidden neurons, and one output neuron. Let W be a weight matrix of size $d \times h$, and $(b_j)_{j=1}^h$ a collection of biases. We denote by $W_{i,:}$ the ith row of the weight matrix and by $W_{:,j}$ its jth column. The neural network computes:

$$\begin{aligned} a_j &= \max(0, W_{:,j}^\top \boldsymbol{x} + b_j) && \text{(layer 1)} \\ f(\boldsymbol{x}) &= \textstyle\sum_j s_j a_j && \text{(layer 2)} \end{aligned}$$

where $s_j \in \{-1, 1\}$ are fixed parameters. The first layer detects patterns of the input data, and the second layer computes a fixed linear combination of these detected patterns.

(a) *Show* that the gradient norm of the network can be upper-bounded as:

$$\Big\|\frac{\partial f}{\partial \boldsymbol{x}}\Big\| \leq \sqrt{h} \cdot \|W\|_F$$

(b) Let $\|W\|_{\text{Mix}} = \sqrt{\sum_i \|W_{i,:}\|_1^2}$ be a ℓ_1/ℓ_2 mixed matrix norm. *Show* that the gradient norm of the network can be upper-bounded by it as:

$$\Big\|\frac{\partial f}{\partial \boldsymbol{x}}\Big\| \leq \|W\|_{\text{Mix}}$$

(c) *Show* that the mixed norm provides a bound that is tighter than the one based on the Frobenius norm, i.e. show that:

$$\|W\|_{\text{Mix}} \leq \sqrt{h} \cdot \|W\|_F$$

.

Exercise 3: Programming (40 P)

Download the programming files on ISIS and follow the instructions.