STATISTICAL MACHINE LEARNING (WS2023/24)
EMPIRICAL RISK MINIMIZATION

Assignment 1. Let $\mathcal{X} = [a, b] \subset \mathbb{R}$, $\mathcal{Y} = \{+1, -1\}$, $\ell(y, y') = [\![y \neq y']\!]$, $p(x \mid y = +1) = p(x \mid y = -1)$ be uniform distributions on $\mathcal{X}$ and $p(y = +1) = 0.8$. Consider the learning algorithm which for a given training set $\mathcal{T}^m = \{(x^1, y^1), \ldots, (x^m, y^m)\}$ returns the strategy

$$h_m(x) = \begin{cases} y^j & \text{if } x = x^j \text{ for some } j \in \{1, \ldots, m\} \\ -1 & \text{otherwise} \end{cases}$$

a) Show that the training error $R_{\mathcal{T}^m}(h_m) = \frac{1}{m} \sum_{i=1}^{m} \ell(y^i, h_m(x^i))$ equals 0 with probability 1 for any finite m.

b) Show that the expected risk $R(h_m) = \mathbb{E}_{(x,y)\sim p}(\ell(y, h_m(x))$ equals 0.8 with probability 1 for any finite m.

c) Give an example of another learning algorithm which returns a strategy h_m whose training error $R_{\mathcal{T}^m}(h_m)$ equals 0 with probability 1 for any finite m.

Assignment 2. The ULLN for finite hypothesis space $\mathcal{H} = \{h_1, \ldots, h_K\}$ ensures that for every $\varepsilon > 0$ we have that

$$\mathbb{P}\Big(\max_{h \in \mathcal{H}} |R_{\mathcal{T}^m}(h) - R(h)| \geq \varepsilon \Big) \leq 2|\mathcal{H}| e^{-\frac{2m\,\varepsilon^2}{(\ell_{\max} - \ell_{\min})^2}} , \tag{1}$$

where $\mathcal{T}^m$ is a training set i.i.d. generated from an arbitrary $p(x, y)$. Use (1) to prove validity of the generalization bound which states that for every $\delta \in [0, 1]$ the upper bound

$$R(h) \leq R_{\mathcal{T}^m}(h) + (\ell_{\max} - \ell_{\min}) \sqrt{\frac{\log(2|\mathcal{H}|) + \log \frac{1}{\delta}}{2m}}$$

holds true for all $h \in \mathcal{H}$ simultaneously with probability $1 - \delta$ at least.

Assignment 3. Assume we train a Convolution Neural Network (CNN) classifier $h \colon \mathcal{X} \to \mathcal{Y}$ to predict a digit $y \in \mathcal{Y} = \{0, 1, \ldots, 9\}$ from an image $x \in \mathcal{X}$. We train the CNN by the Stochastic Gradient Descent (SGD) algorithm using 100 epochs. After each epoch we save the current weights so that at the end of training we have a set $\mathcal{H} = \{h_i \colon \mathcal{X} \to \mathcal{Y} \mid i = 1, \ldots, 100\}$ containing 100 different CNN classifiers. We select the best CNN classifier $\hat{h}$ out of $\mathcal{H}$ based on the validation error $R_{\mathcal{V}^m}(h)$, i.e. $\hat{h} = \arg\min_{h \in \mathcal{H}} R_{\mathcal{V}^m}(h)$, where

$$R_{\mathcal{V}^m}(h) = \frac{1}{m} \sum_{i=1}^{m} [\![y^j \neq h(x^j)]\!] ,$$

is computed from the validation set $\mathcal{V}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \ldots, m\}$ i.i.d. drawn from $p(x, y)$. The validation set $\mathcal{V}^m$ has not been used by SGD for training. We are left with no additional examples for testing and hence we use the validation error $R_{\mathcal{V}^m}(\hat{h})$ also as an estimate of the generalization error $R(\hat{h}) = \mathbb{E}[\ell(y, \hat{h}(x)]$.

a) Can you quantify how well the validation error $R_{\mathcal{V}^m}(\hat{h})$ estimates the generalization error $R(\hat{h})$?

b) What is the minimum number of examples in the validation set $\mathcal{V}^m$ required to confidently assert, with a 99% probability, that the generalization error $R(\hat{h})$ deviates from the validation error $R_{\mathcal{V}^m}(\hat{h})$ by no more than 1% ?