DEEP LEARNING (SS2022)
SEMINAR 2

Assignment 1 (Chebyshev). Let X be a real valued random variable with expectation $\mathbb{E}X$ and finite variance $\mathbb{V}X$. The Chebyshev inequality asserts

$$\mathbb{P}\big(|X - \mathbb{E}X| > \varepsilon\big) \leqslant \frac{\mathbb{V}X}{\varepsilon^2}.$$

Let X_i, $i = 1, \ldots, m$ be independent, identically distributed random variables with expectation $\mathbb{E}X$ and finite variance $\mathbb{V}X$ and let $Y = \frac{1}{m}\sum_{i=1}^{m} X_i$ be their empirical mean. Prove the inequality

$$\mathbb{P}\big(|Y - \mathbb{E}Y| > \varepsilon\big) \leqslant \frac{\mathbb{V}X}{m\varepsilon^2}.$$

Assignment 2 (Hoeffding). Let X_i, $i = 1, \ldots, m$ be independent random variables bounded by the interval $[a, b]$, i.e. $a \leqslant X_i \leqslant b$. Let $X = \frac{1}{m}\sum_{i=1}^{m} X_i$ be their empirical mean. The Hoeffding inequality asserts that

$$\mathbb{P}\big(|X - \mathbb{E}X| > \varepsilon\big) \leqslant 2\exp\left(-\frac{2m\varepsilon^2}{(b-a)^2}\right).$$

Let us now consider a predictor $h\colon \mathcal{X} \to \mathcal{Y}$, and a loss $\ell(y, y')$. The risk of the predictor is denoted by $R(h)$ and its empirical risk on a test set $\mathcal{T}^m = \big\{(x^j, y^j) \;\big|\; j = 1, \ldots, m\big\}$ is denoted by $R_{\mathcal{T}^m}(h)$.

a) Prove that the generalisation error of h can be bounded in probability by

$$\mathbb{P}\Big(|R(h) - R_{\mathcal{T}^m}(h)| > \varepsilon\Big) < 2e^{-\frac{2m\varepsilon^2}{(\triangle\ell)^2}}, \tag{1}$$

where $\triangle\ell = \ell_{max} - \ell_{min}$.

b) Verify the value m given in Example 1. of Lecture 2. for the special case of a binary classifier and the 0/1-loss.

c*) We want to utilise the Hoeffding inequality for choosing the best predictor from a finite set of predictors $\mathcal{H}$. Denoting the r.h.s. of (1) by δ, we interpret it as follows. Among all possible test sets $\mathcal{T}^m$ of size m there are at most $\delta * 100$ percent "bad" test sets for a given predictor h. We call a test set $\mathcal{T}^m$ bad for the predictor h if $|R(h) - R_{\mathcal{T}^m}(h)| > \varepsilon$. Conclude that the percentage of test sets, which are bad for at least one $h \in \mathcal{H}$ can be bounded by

$$\mathbb{P}\Big(\max_{h\in\mathcal{H}}|R(h) - R_{\mathcal{T}^m}(h)| > \varepsilon\Big) < 2|\mathcal{H}|e^{-\frac{2m\varepsilon^2}{(\triangle\ell)^2}}$$

Assignment 3 (Log Softmax)**.** Consider a neural network with outputs y_k, $k = 1, \dots, K$ representing posterior class probabilities. The last layer of this network is a softmax layer with output

$$y_k = \frac{e^{x_k}}{\sum_\ell e^{x_\ell}},$$

where x_k are the outputs of the last linear layer and represent class scores. When learning such a network by maximising the log conditional likelihood, we have to consider log-probabilities

$$z_k = \log y_k = x_k - \log \sum_\ell e^{x_\ell}$$

We will analyse the nonlinear part of the r.h.s.

$$f(x) = \log \sum_\ell e^{x_\ell}$$

a) Prove that its gradient is given by $\nabla f(x) = y$, i.e. by the vector of class probabilities. Conclude that the norm of the gradient is bounded by 1.

b*) Compute the second derivative of f and show that it can be expressed as

$$\nabla^2 f(x) = \mathrm{Diag}(y) - yy^T.$$

Prove that this matrix is positive semi-definite and conclude that $f(x)$ is a convex function. Note that the second derivative is the Jacobian of softmax.

Assignment 4 (Backprop)**.** Given an operation with the output y and the derivative of the loss w.r.t. y – a row vector J_y, the "backprop" operation needs to compute derivatives w.r.t. all inputs. Compute the backprop of the following operations:

a) $y = |x|$, where the absolute value is applied coordinate-wise to a vector x.

b) $y = x + z$

c) $y = (x; z)$ — the concatenated vector of x and z

d) Convolution in 1D: $y_i = \sum_k w_k x_{i+k} + b_i$. The inputs are: w, x, b. For simplicity, do not infer index ranges.

Assignment 5 (Backprop of Scan)**.** In Adaboost classifiers, a commonly used feature is the difference of average brightness in two rectangles in the image. The average over arbitrary rectangle can be computed very cheaply if the so-called *integral image* (AKA *cumulative sum*, *scan*) is precomputed. In this exercise we want to make this operation differentiable.
The *inclusive cumulative sum* operation (in 1D) is defined as follows. Given the input vector $x \in \mathbb{R}^n$ the output $y \in \mathbb{R}^n$ has components:

$$y_i = \sum_{j \le i} x_j.$$

Compute the backprop of scan, i.e. given the derivative J_y, compute the derivative J_x.