DEEP LEARNING (SS2024)
SEMINAR 3

Assignment 1 (Sampling with Replacement)**.** In deep learning examples forming mini-batches are drawn randomly *without* replacement. Strictly speaking, this approach is not i.i.d. because the realization of a mini-batch depends on the realizations of the previous mini-batches. The scheme has however proven to be much more efficient for learning in practice. We will inspect theoretical reasons for this by analyzing the i.i.d. sampling strategy with replacement.

a) Let the dataset contain n examples. During an epoch, we make n random draws *with* replacement. What is the probability that a particular example i will *not* be drawn in the epoch? What is the limit of this probability for $n \to \infty$?

Hint: Use L'Hôpital's rule to compute the limit.
(or compute it *e.g.* with `www.wolframalpha.com`)

b) We conclude that a considerable portion of training data will remain unused in a given epoch when sampling *with* replacement. It is therefore natural to ask the following question: What is the expected number of epochs we need to run in order to have each example being drawn at least once? This question corresponds to the "Coupon collector's problem" (see Wikipedia), Establish a relation to this problem and use the formula from there to find the expected time.

Assignment 2 (EWA and Momentum)**.**

a) PyTorch defines SGD with momentum as follows:

$$\begin{aligned} v_{t+1} &= \mu v_t + \tilde{g}_t \\ \theta_{t+1} &= \theta_t - \varepsilon v_{t+1}, \end{aligned} \tag{1}$$

where $\tilde{g}_t$ is the stochastic gradient at the point θ_t. Derive this algorithm by applying EWA to stochastic gradient estimates in plain SGD (SGD lecture slides 13 and 14). How is the momentum parameter μ related to q in EWA?

b) The initial value of v_0 may have an undesirable effect in the beginning of training, before its weight becomes negligibly small (weight w_0 in SGD lecture slide 14). We address this problem by designing the coefficients q smoothly transiting from flat mean to exponentially weighted mean and we will verify that it is equivalent to momentum in Adam optimizer.

Let $\tilde{g}_t$ for $t = 1, \ldots n$ be a sequence of stochastic gradients obtained for the same model parameter vector θ by sampling mini-batches at random without doing any optimization steps (or doing the optimization, but with infinitesimally small step size). Consider the following variant of the exponentially weighted average:

$$v_t = (1 - q_t)v_{t-1} + q_t \tilde{g}_t, \tag{2}$$

where $v_0 = 0$, $q_t = \frac{q}{1-(1-q)^t}$ and q is a constant.

Show that for any $t \geq 1$, v_t is an unbiased estimator of the true gradient.
Hint: Start by showing it for $t = 1$ and $t = 2$.

c) Alternatively, consider the usual EWA with constant q:

$$\hat{v}_t = (1 - q)\hat{v}_{t-1} + q\tilde{g}_t \tag{3}$$

and show that $\bar{v}_t = \hat{v}_t/(1 - (1 - q)^t)$ is also unbiased and coincides with v_t.
Hint: Use the mathematical induction over t to prove the equivalence.

d) Inspect the Adam optimizer in PyTorch and the implementation of momentum there (consider only the first order moment, the relevant momentum parameter is β_1). Which of the above EWA methods is being used, the one derived in **b)** or in **c)** ?

Assignment 3 (Receptive fields)**.** Consider a convolutional network consisting of convolution layers and max-pooling layers. Each of them is characterized by a kernel size k_ℓ and a stride s_ℓ. The *receptive field* of a neuron in layer ℓ is the bounding box of all nodes in the input layer that can influence its output. Let us define the *stride of the receptive field* as the shift in pixels between receptive fields of two neighboring neurons in layer ℓ. Knowing the receptive field size S_ℓ and receptive field stride T_ℓ of neurons in layer ℓ and the kernel size k and stride s of the next operation (convolution or max pooling), find the receptive field size and stride of neurons in layer $\ell + 1$.

Note: This relation will be needed for the lab on CNN visualization & adversarial patterns.

N.B. The *effective receptive field*, the set of pixels in the input which have a non-negligible average contribution to the neuron's activation, depends also on the network weights and is usually substantially smaller.

Assignment 4 (SGD step)**.** Let $f(\theta)$ denote the loss function and let $g^t = \nabla_\theta f(\theta^t)$ denote the gradient at θ^t. The standard gradient descent step is $\theta^{t+1} = \theta^t - \varepsilon g^t$. Show that the step vector $\Delta\theta = -\varepsilon g^t$ is the solution to the following optimization problem:

$$\underbrace{f(\theta^t) + \langle \Delta\theta, g^t \rangle}_{\text{Approximation of } f(\theta + \Delta\theta)} + \underbrace{\frac{1}{2\varepsilon}\|\Delta\theta\|_2^2}_{\text{Penalty for step length}} \to \min_{\Delta\theta}. \tag{4}$$

This expression occurred when we discussed implicit regularization of SGD (slide 20) and will be also used in the adaptive methods.