DEEP LEARNING (SS2023)
SEMINAR 3

Assignment 1 (Sampling with Replacement)**.** In deep learning examples forming mini-batches are drawn randomly without replacement. This ensures that every example in the dataset is used exactly once in each epoch. Strictly speaking, this approach is not i.i.d. because the realisation of a mini-batch depends on the realisations 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 be not 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. 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:

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

where 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 slide 13). 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 12). We will design coefficients q smoothly transiting from flat mean to exponentially weighted mean, addressing this problem and will verify that it is equivalent to momentum in Adam optimizer.

Let g_t for $t = 1, \dots n$ 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. Consider the following variant of the exponentially weighted average:

$$v_t = (1 - q_t)v_{t-1} + q_t g_t,$$

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

1. Show that for any $t \geq 1$, v_t is an unbiased estimator of the true gradient. *Hint:* start by showing it for $t = 1$, $t = 2$.
2. Consider the usual EWA with constant q:

$$\hat{v}_t = (1 - q)\hat{v}_{t-1} + qg_t.$$

   Show that $\hat{v}_t/(1 - (1 - q)^t)$ is also unbiased and coincides with v_t.
3. Inspect the Adam optimizer in Pytorch and the implementation of momentum there (the relevant momentum parameter is beta1). Which EWA method is being used?

Assignment 3 (Receptive fields)**.** The receptive field of a hidden layer neuron is defined as the set of all pixels in the input image, which potentially influence the neuron's activation (assuming generic weights).

What is the size of the receptive field of an output unit of the following fully convolutional network:

$$\begin{gathered}\text{conv(5}\times\text{5, stride 1, dilation 1)}\\ \text{conv(3}\times\text{3, stride 1, dilation 2)}\\ \text{conv(3}\times\text{3 stride 2, dilation 1)}\\ \text{MaxPool(2,2)}\end{gathered}$$

where dilation 1 means standard convolution without "holes" and dilation 2 is as illustrated in the CNN lecture slides.

N.B. The *effective receptive* field of pixels in the input, which have a non-negligible average contribution to the neuron's activation, depends 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:

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

This expression occurred when we discussed implicit regularization of SGD (slide 17) and will be also used in the adaptive methods.