Deep Learning: Assignments with Solutions

Assignment 1 (Node statistics)**.** Let us consider a neuron in a linear layer of a classification network. Its output is given by

$$y = \sum_{i=1}^{n} w_i x_i,$$

where x is the output of the preceding layer. Let us consider the statistics of x over the training data and assume that the components x_i are statistically independent and identically distributed with zero mean and variance σ^2. The weight components w_i are initialized i.i.d. with zero mean and variance $\widetilde{\sigma}^2$. Compute the mean and variance of y.
Solution. Since w_i and x_i are statistically independent, we obtain the mean of y by

$$\mathbb{E}[y] = \sum_{i=1}^{n} \mathbb{E}[w_i]\mathbb{E}[x_i] = 0. \tag{1}$$

To compute the variance of y, we use that $\mathbb{V}[XY] = \mathbb{V}[X]\mathbb{V}[Y]$ and $\mathbb{V}[X+Y] = \mathbb{V}[X] + \mathbb{V}[Y]$ hold for any pair of statistically independent random variables X and Y. We obtain

$$\mathbb{V}[y] = \sum_{i=1}^{n} \mathbb{V}[w_i]\mathbb{V}[x_i] = n\widetilde{\sigma}^2\sigma^2. \tag{2}$$

Assignment 2 (Backpropagation)**.**
Let $x \in \mathbb{R}^N$ be a vector with components x_i for $i = 1, \ldots N$ and consider a layer performing the following computation:

$$y_i = a(x_i + x_{i+2}) + b \quad \text{for } i = 1 \ldots N-2. \tag{3}$$

Given the gradient of the loss function in y, $g := \nabla_y L \in \mathbb{R}^{N-2}$, compute the gradient of the loss in a, b and x.
Solution.

$$\frac{\mathrm{d}L}{\mathrm{d}b} = \sum_{i=1}^{N-2} \frac{\mathrm{d}L}{\mathrm{d}y_i}\frac{\partial y_i}{\partial b} = \sum_{i=1}^{N-2} \frac{\partial L}{\partial y_i} = \sum_{i=1}^{N-2} g_i. \tag{4}$$

$$\frac{\mathrm{d}L}{\mathrm{d}a} = \sum_{i=1}^{N-2} \frac{\mathrm{d}L}{\mathrm{d}y_i}\frac{\partial y_i}{\partial a} = \sum_{i=1}^{N-2} g_i(x_i + x_{i+2}). \tag{5}$$

$$\frac{\mathrm{d}L}{\mathrm{d}x_j} = \sum_{i=1}^{N-2} g_i \frac{\partial y_i}{\partial x_j} = \sum_{i=1}^{N-2} g_i a\big([\![j{=}i]\!] + [\![j{=}i{+}2]\!]\big) = \begin{cases} a g_j & \text{if } j \leq 2, \\ a(g_j + g_{j-2}) & \text{if } j = 2, \dots N-2, \\ a g_{j-2} & \text{if } j \geq N-2. \end{cases} \tag{6}$$

Assignment 3 (SGD with Regularization)**.**
Consider a regularized loss function $\tilde{L}(\theta) = L(\theta) + \frac{\lambda}{2}\|\theta\|^2$. Let θ^t be the current parameter estimate and g^t be the gradient of L at θ^t.

a) Give an update step for an SGD-like algorithm that applies a variance reduction technique to stochastic gradients g^t in order to obtain smoothed estimates $\tilde{g}_t$.

b) Solve the following proximal step problem

$$\theta^{t+1} = \arg\min_\theta \Big[\big\langle \tilde{g}^t, \theta - \theta^t \big\rangle + \frac{\lambda}{2}\|\theta\|^2 + \frac{1}{2\varepsilon'}\|\theta - \theta^t\|^2\Big]. \tag{7}$$

Solution. **a)** To reduce the variance of stochastic gradients g^t we will use exponentially weighted average with parameter q.

$$\tilde{g}^t := \tilde{g}^{t-1}(1-q) + g^t q. \tag{8}$$

Then we write standard SGD step using the gradient $\tilde{g}^t + \lambda\theta^t$ — the smoothed gradient of L plus the gradient of regularization at θ^t:

$$\theta^{t+1} := \theta^t - \varepsilon(\tilde{g}^t + \lambda\theta^t). \tag{9}$$

b)

$$\theta^{t+1} = \arg\min_\theta \Big[\big\langle \tilde{g}^t, \theta - \theta^t \big\rangle + \frac{\lambda}{2}\|\theta\|^2 + \frac{1}{2\varepsilon'}\|\theta - \theta^t\|^2\Big]. \tag{10}$$

Solving for stationary point:

$$0 = \frac{\mathrm{d}}{\mathrm{d}\theta} = \tilde{g}^t + \lambda\theta + \frac{1}{\varepsilon'}(\theta - \theta^t). \tag{11}$$

We find:

$$\theta^{t+1} = \frac{\theta^t - \varepsilon'\tilde{g}^t}{\varepsilon'\lambda + 1}. \tag{12}$$

Remark. We can check that by setting $\varepsilon' = \frac{\varepsilon}{1-\lambda\varepsilon}$ this solution matches the common SGD step (9), *i.e.* for quadratic regularization linearizing it or considering explicitly in the proximal problem is equivalent.

Assignment 4 (Adversarial attack)**.** Let us consider a neural network for classification with predictive class log probabilities given by the vector $f(x;\theta) \in \mathbb{R}^K$. An attacker wants to find a perturbed image $\tilde{x}$ satisfying $|\tilde{x}_i - x_i| < \varepsilon$ for all i such that it would minimize the probability of predicting the correct label y.
Formulate the attacker's task as an optimization problem using a *linear approximation* of f in the box $|\tilde{x}_i - x_i| < \varepsilon$. Solve this problem.

Solution. The log probability of the correct label is $f_y(x;\theta)$ and its linear approximation in the neighbourhod of x is given by

$$f_y(\tilde{x};\theta) \approx f_y(x;\theta) + g^T(\tilde{x} - x), \tag{13}$$

where g denotes the gradient $\nabla_x f_y(x;\theta)$. The attackers task is

$$g^T(\tilde{x} - x) \to \min_{\tilde{x}} \tag{14}$$

$$\text{s.t.} \quad |\tilde{x}_i - x_i| < \varepsilon \quad \forall i. \tag{15}$$

It decomposes into independent tasks for each $\tilde{x}_i$ with solution $\tilde{x}_i^* = -\varepsilon \operatorname{sign}(g_i)$.

Assignment 5 (KL divergence and cross entropy)**.**
Assume that the training data are given by a generator $p^*(y,x)$. We want to learn the conditional distribution $p(y \,|\, x;\theta)$ in the form of a neural network parametrized by θ. Prove that minimizing $\mathbb{E}_{p^*(x)}[D_{\mathrm{KL}}(p^*(y \,|\, x) \,\|\, p(y \,|\, x;\theta))]$ is equivalent to minimizing the expected cross-entropy of $p(y \,|\, x;\theta)$ relative to $p^*(y \,|\, x)$, where the expectation is taken over $p^*(x)$.

Solution. Let us expand the KL divergence for a give x:

$$D_{\mathrm{KL}}(p^*(y \,|\, x) \,\|\, p(y \,|\, x;\theta)) = \int_y p^*(y \,|\, x) \log \frac{p^*(y \,|\, x)}{p(y \,|\, x;\theta)} \tag{16}$$

$$= \underbrace{\int_y p^*(y \,|\, x) p^*(y \,|\, x)}_{\text{does not depend on } \theta} \underbrace{- \int_y p^*(y \,|\, x) \log p(y \,|\, x;\theta)}_{\text{cross-entropy}}. \tag{17}$$

Taking expectation in $p^*(x)$ the first term still does not depend on θ and thus optimization with or without it is equivalent.

Assignment 6 (SGD with Regularization 2)**.**
Consider a regularized loss function $\tilde{L}(\theta) = L(\theta) + \rho(\|\theta\|)$, where $\rho\colon \mathbb{R}_+ \to \mathbb{R}_+$ is a differentiable function. Let θ^t be the current parameter estimate and g be the gradient of L at θ^t. Show that the solution of the composite proximal step problem

$$\arg\min_{\theta}\Big[\big\langle g, \theta - \theta^t\big\rangle + \rho(\|\theta\|) + \frac{1}{2\varepsilon}\|\theta - \theta^t\|^2\Big] \tag{18}$$

for a sufficiently small ε takes the form: $\theta = \frac{a}{\|a\|}l$, where $a = \theta^t - \varepsilon g$ is the usual non-regularized SGD update and and l is a root of the equation $l + \varepsilon\rho'(l) = \|a\|$.
Solution. We solve for a critical point:

$$\begin{aligned}0 = \frac{\partial}{\partial\theta} &= g + \rho'(\|\theta\|)\frac{\theta}{\|\theta\|} + \frac{1}{\varepsilon}(\theta - \theta^t)\\ \theta(\frac{\varepsilon\rho'(\|\theta\|)}{\|\theta\|} + 1) &= \theta^t - \varepsilon g.\end{aligned} \tag{19}$$

Since $(\frac{\varepsilon\rho'(\|\theta\|)}{\|\theta\|} + 1)$ is a scalar we conclude that θ will be proportional to $\theta^t - \varepsilon g =: a$. Take the norm of the vectors on both sides in (19):

$$\begin{aligned}\|\theta\|\Big(\frac{\varepsilon\rho'(\|\theta\|)}{\|\theta\|} + 1\Big) &= \|a\|\\ \varepsilon\rho'(\|\theta\|) + \|\theta\| &= \|a\|.\end{aligned} \tag{20}$$

Denoting $l = \|\theta\|$, we can express $\frac{\varepsilon\rho'(\|\theta\|)}{\|\theta\|} + 1 = \frac{\|a\|}{l}$. The equation (20) holds for ε sufficiently small so that the value of $\frac{\varepsilon\rho'(\|\theta\|)}{\|\theta\|}$ is positive, otherwise its absolute value needs to be taken.

Assignment 7 (Shift of Prior)**.** A neural network with softmax activation in the last layer has been trained for classifying patterns by predicting the posterior class probabilities $p(y \mid x)$, $y \in K$. The relative class frequencies in the training set were $p(y)$. When applying the network, it turned out that the prior class probabilities for real data are different and equal to $p^*(y)$. Explain how to use the network as a predictor without re-training it. We assume the 0/1 loss for prediction.
Solution. Let us denote the distribution of the training data by $p(x, y)$. We have

$$p(x, y) = p(x \mid y)p(y) = p(y \mid x)p(x)$$

and the trained network estimates $p(y \mid x)$. Let us denote the data distribution in the application by $p_a(x, y)$. We have

$$p_a(x, y) = p(x \mid y)p^*(y) = p_a(y \mid x)p_a(x),$$

i.e. $p(x \mid y)$ remains unchanged and $p(y)$ changes to $p^*(y)$. Comparing the two equations we get

$$p_a(y \mid x) \propto \frac{p^*(y)}{p(y)} p(y \mid x).$$

Hence, the trained network can be used in the application just by reweighting its softmax outputs by the factors $\frac{p^*(y)}{p(y)}$ and deciding for the class with the largest reweighted output.

Assignment 8 (K-means)**.** Let us consider the standard *k-means clustering* problem for data $x \in \mathbb{R}^n$ and K cluster centers $y_k \in \mathbb{R}^n$

$$\sum_{x \in \mathcal{T}^m} \min_k \|x - y_k\|^2 \to \min_y,$$

where $y = (y_1, \ldots, y_K)$ denotes the set of all cluster centers and $\mathcal{T}^m$ denotes the training set.

a) Propose a stochastic gradient descent method that operates in full online mode. I.e. it receives *one* example per iteration (the mini-batch size is 1). Explain why it is necessary to choose a decreasing learning rate.

b) What is the run-time complexity for a training epoch? Compare it with the run-time complexity of the standard k-means algorithm.

Solution. **a)** Given a single training example $x \in \mathcal{T}^m$, we have the objective $f(y) = \min_k \|x - y_k\|^2$ and its gradients w.r.t. the cluster centers are

$$\nabla_{y_k} f(y) = \begin{cases} 2(y_k - x) & \text{if } k = \arg\min_{k'} \|x - y_{k'}\|^2, \\ 0 & \text{otherwise.} \end{cases}$$

We obtain the following SGD algorithm for the problem.
Given a training example x do
(1) find the closest cluster center $k = \arg\min_{k'} \|x - y_{k'}\|^2$,
(2) update $y_k \to y_k + \alpha(t)(x - y_k)$,
where $\alpha(t)$ is a decreasing learning rate. The algorithm will not converge to a local minimum if the learning rate is constant. Instead, it will keep oszillating around it.

b) The run-time complexity of the SGD algorithm for one training epoch is $\mathcal{O}(nmK)$. The standard k-means algorithm iteration consists of two steps (i) assignment and (ii) update. The run-time complexity of the former dominates and is $\mathcal{O}(nmK)$.

Assignment 9 (Backprop)**.**
Let $x \in \mathbb{R}^n$. Consider the following normalized linear layer:

$$y_i = \frac{w_i^\mathsf{T} x + b_i}{\|w_i\|},$$

where $w_i \in \mathbb{R}^n$ for $i = 1 \ldots m$, $b_i \in \mathbb{R}$ and $\|w_i\|$ is the Euclidean norm of vector w_i. Given the gradient of the loss function in y, $g := \nabla_y L \in \mathbb{R}^m$, compute gradients of the loss in w, b, x.

Solution. We will use general the total derivative rule

$$\frac{\mathrm{d}L}{\mathrm{d}\theta} = \sum_i \frac{\mathrm{d}L}{\mathrm{d}y_i} \frac{\partial y_i}{\partial \theta} = \sum_i g_i \frac{\partial y_i}{\partial \theta}. \tag{21}$$

Since y_i depends only on b_i and not on b_j for $j \neq i$ for $\nabla_b L$ we have

$$\frac{\mathrm{d}L}{\mathrm{d}b_i} = g_i \frac{\partial y_i}{\partial b_i} = \frac{g_j}{\|w_j\|}. \tag{22}$$

For $\nabla_x L$ we have

$$\frac{\mathrm{d}L}{\mathrm{d}x_j} = \sum_i g_i \frac{\partial y_i}{\partial x_j} = \sum_i g_i \frac{w_{ij}}{\|w_i\|}. \tag{23}$$

Since y_i depends only on w_i and not on w_j for $j \neq i$ for $\nabla_w L$ we have

$$\frac{\mathrm{d}L}{\mathrm{d}w_i} = \sum_i g_i \frac{\partial y_i}{\partial w_i} = \sum_i g_i \Big(\frac{x}{\|w_i\|} + (w_i^\mathsf{T} x + b_i) \frac{-w_i}{\|w_i\|^3} \Big). \tag{24}$$

Assignment 10 (VAE)**.**
Consider a variational autoencoder with the decoder model being a normal distribution $p(x|z) = \mathcal{N}(x; \mu(z), \sigma^2 I)$, where $x \in \mathbb{R}^d$ and σ is a parameter. Show that the optimal value of the variance σ^2 for the evidence lower bound

$$\text{ELBO} = \mathbb{E}_{p_d(x)} \mathbb{E}_{q(z|x)} \big[\log p(x|z) \big] - D_{KL}(q(z|x) \parallel p(z))$$

with the current encoder $q(z|x)$ is given by

$$\sigma^2 = \frac{1}{d} \mathbb{E}_{p_d(x)} \mathbb{E}_{q(z|x)} \big[\|x - \mu(z)\|^2 \big].$$

Solution. The density of the Normal distribution with diagonal covariance matrix is

$$p(x|z) = \Big(\frac{1}{\sqrt{2\pi}\sigma}\Big)^d \exp(-\frac{\|x-\mu(z)\|^2}{2\sigma^2}). \tag{25}$$

Respectively the log density is

$$\log p(x|z) = \log\Big(\frac{1}{\sqrt{2\pi}\sigma}\Big)^d - \frac{\|x-\mu(z)\|^2}{2\sigma^2} = -\frac{d}{2}d\log(2\pi) - d\log\sigma - \frac{\|x-\mu(z)\|^2}{2\sigma^2}. \tag{26}$$

Note that the log density is a convex function of σ. We find optimum by finding stationary points of ELBO in σ. The KL divergence term does not depend on σ and its derivative is zero. Since the expectation densities do not depend on σ, the derivative can be interchanged with expectation:

$$\frac{\partial}{\partial\sigma}\mathrm{ELBO} = \mathbb{E}_{p_d(x)}\mathbb{E}_{q(z|x)}\big[\frac{\partial}{\partial\sigma}\log p(x|z)\big]. \tag{27}$$

We then calculate

$$\frac{\partial}{\partial\sigma}\log p(x|z) = -\frac{d}{\sigma} + \frac{\|x-\mu(z)\|^2}{\sigma^3}. \tag{28}$$

And solve

$$\mathbb{E}_{p_d(x)}\mathbb{E}_{q(z|x)}\big[-\frac{d}{\sigma} + \frac{\|x-\mu(z)\|^2}{\sigma^3}\big] = 0. \tag{29a}$$

$$\frac{d}{\sigma} = \frac{1}{\sigma^3}\mathbb{E}_{p_d(x)}\mathbb{E}_{q(z|x)}\big[\|x-\mu(z)\|^2\big]. \tag{29b}$$

$$\sigma^2 = \frac{1}{d}\mathbb{E}_{p_d(x)}\mathbb{E}_{q(z|x)}\big[\|x-\mu(z)\|^2\big]. \tag{29c}$$

Remark. The solution takes the same form as the maximum likelihood estimate of variance from supervised data samples x, z. The difference is that here we do not know the ground truth samples (x, z) and estimate them using the current encoder, *i.e.*, draw them from the distribution $p_d(x)q(z|x)$.

Assignment 11 (Mirror Descent)**.**
Solve the proximal step problem:

$$\min_x \langle\nabla f(x^0), x - x^0\rangle + \frac{1}{\varepsilon}D(x, x^0),$$

where $x^0 \in (0, 1)$ and

$$D(x, x^0) = \sum_i (x_i \log\frac{x_i}{x_i^0} + (1-x_i)\log\frac{1-x_i}{1-x_i^0}).$$

Hint: The problem is convex and can be solved by stationary point conditions.

Solution. The objective is a sum of terms where each summand i depends on x_i only. Therefore minimization decouples into independent minimizations over x_i:

$$\min_{x_i} \langle g_i, x_i - x_i^0 \rangle + \frac{1}{\varepsilon}(x_i \log \frac{x_i}{x_i^0} + (1 - x_i) \log \frac{1 - x_i}{1 - x_i^0}),$$

where $g = \nabla f(x^0)$. We solve for the critical point x_i:

$$\begin{aligned}
0 &= \frac{\partial}{\partial x_i} = g_i + \frac{1}{\varepsilon}(\log \frac{x_i}{x_i^0} - \log \frac{1 - x_i}{1 - x_i^0}) \\
0 &= -\varepsilon g_i + \log \frac{x_i}{1 - x_i} - \log \frac{x_i^0}{1 - x_i^0} \\
\log \frac{x_i}{1 - x_i} &= \log \frac{x_i^0}{1 - x_i^0} - \varepsilon g_i \\
x_i &= \text{sigmoid}\Big(\log \frac{x_i^0}{1 - x_i^0} - \varepsilon g_i \Big).
\end{aligned}$$

Remark. Suppose we solve these proximal problems iteratively and x^t is the current iteration. Denote $y^t = \operatorname{logit}(x^t) = \log \frac{x^t}{1-x^t}$, then $x^t = \operatorname{sigmoid}(y^t)$ and on the next iteration we do not need to calculate $\log \frac{x^t}{1-x^t}$, we could just reuse y^t. Then the iterates can be simplified to

$$\begin{aligned}
y^{t+1} &= y^t - \varepsilon g, \\
x^{t+1} &= \operatorname{sigmoid}(y^{t+1}).
\end{aligned}$$