MATH3431 Machine Learning and Neural Networks III Epiphany term 2023

Exercise sheet

Lecturer: Georgios P. Karagiannis georgios.karagiannis@durham.ac.uk

Part 1. Stochastic learning

Exercise 1. ($\star$)Let $f:\mathbb{R}^d\rightarrow\mathbb{R}$ such that $f\left(w\right)=g\left(<w,x>+y\right)$ or some $x\in\mathbb{R}^d$, $y\in\mathbb{R}$. Show that: If g is convex function then $f$is convex function.

Solution. Let $u,v\in\mathbb{R}^d$ and $a\in[0,1]$. It is

$$\begin{aligned}
f\left(\alpha u+\left(1-\alpha\right)v\right)= & g\left(<\alpha u+\left(1-\alpha\right)v,x>+y\right)\\
= & g\left(<\alpha u,x>+<\left(1-\alpha\right)v,x>+y\right)\\
= & g\left(\alpha\left(<u,x>+y\right)+\left(1-\alpha\right)\left(<v,x>+y\right)\right) && y=\alpha y+\left(1-\alpha\right)y\\
\le & \alpha g\left(<u,x>+y\right)+\left(1-\alpha\right)g\left(<v,x>+y\right) && (g\text{ is convex})\\
= & \alpha f\left(u\right)+\left(1-\alpha\right)f\left(v\right)
\end{aligned}$$

Exercise 2. ($\star$)Let functions g_1 be ρ_1-Lipschitz and g_2 be ρ_2-Lipschitz. Then, show that, f with $f\left(x\right)=g_1\left(g_2\left(x\right)\right)$ is $\rho_1\rho_2$-Lipschitz.

Solution.

$$\begin{aligned}
\left|f\left(w_1\right)-f\left(w_2\right)\right|= & \left|g_1\left(g_2\left(w_1\right)\right)-g_1\left(g_2\left(w_2\right)\right)\right|\\
\le & \rho_1\left|g_2\left(w_1\right)-g_2\left(w_2\right)\right|\\
\le & \rho_1\rho_2\left|w_1-w_2\right|
\end{aligned}$$

Exercise 3. ($\star$)Let $f:\mathbb{R}^d\rightarrow\mathbb{R}$ with $f\left(w\right)=g\left(\langle w,x\rangle+y\right)$ $x\in\mathbb{R}^d$ and $y\in\mathbb{R}$. Let $g:\mathbb{R}\rightarrow\mathbb{R}$ be a β-smooth function. Then show that f is a $\left(\beta\left\Vert x\right\Vert^2\right)$-smooth.

Hint:: You may use Cauchy-Schwarz inequality $\langle y,x\rangle\le\left\Vert y\right\Vert\left\Vert x\right\Vert$

$$
\begin{aligned}
f(v) =& g(\langle w, x\rangle + y) \\
\leq& g(\langle w, x\rangle + y) + g'(\langle w, x\rangle + y)\langle v - w, x\rangle + \frac{\beta}{2}(\langle v - w, x\rangle)^2 && (g \text{ is smooth}) \\
\leq& g(\langle w, x\rangle + y) + g'(\langle w, x\rangle + y)\langle v - w, x\rangle + \frac{\beta}{2}(\|v - w\| \|x\|)^2 && (\text{Cauchy-Schwatz inequality}) \\
=& f(w) + \langle \nabla f(w), v - w\rangle + \frac{\beta \|x\|^2}{2}\|v - w\|^2
\end{aligned}
$$

Exercise 4. ($\star$)Show that $f : S \to \mathbb{R}$ is ρ-Lipschitz over an open convex set S if and only if for all $w \in S$ and $v \in \partial f(w)$ it is $\|v\| \leq \rho$.

Hint:: You may use Cauchy-Schwarz inequality $\langle y, x\rangle \leq \|y\| \|x\|$

Solution. $\Longrightarrow$ Let $f : S \to \mathbb{R}$ be ρ-Lipschitz over convex set S, $w \in S$ and $v \in \partial f(w)$.

- Since S is open we get that there exist $\epsilon > 0$ such as $u := w + \epsilon \frac{v}{\|v\|}$ where $u \in S$. So $\langle u - w, v\rangle = \epsilon \|v\|$ and $\|u - w\| = \epsilon$.
- From the subgradient definition we get
$$f(u) - f(w) \geq \langle u - w, v\rangle = \epsilon \|v\|$$
- From the Lipschitzness of $f(\cdot)$ we get
$$f(u) - f(w) \geq \rho \|u - w\| = \rho\epsilon$$

  Therefore $\|v\| \leq \rho$.

  $\Longleftarrow$ It is for all $w \in S$ and $v \in \partial f(w)$ it is $\|v\| \leq \rho$.
- For any $u \in S$, it is
$$
\begin{aligned}
f(w) - f(u) \leq& \langle v, w - u\rangle && (\text{ because } v \in \partial f(w) \text{ }) \\
\leq& \|v\| \|w - u\| && \text{by Cauchy-Schwarz inequality} \\
\leq& \rho \|w - u\| && \text{because } \|v\| \leq \rho
\end{aligned} \tag{1}
$$
- Similarly it results $u, w \in S$
$$f(w) - f(u) \leq \langle v, u - w\rangle \|v\| \leq \|v\| \|u - w\| \leq \rho \|u - w\|$$

  from (1) because w, u can be swaped in (1) as they both are any values in S.

Exercise 5. ($\star$)Let $g_1(w), ..., g_r(w)$ be r convex functions, and let $f(\cdot) = \max_{\forall j}(g_j(\cdot))$. Show that for some w it is $\nabla g_k(w) \in \partial f(w)$ where $k = \arg\max_j (g_j(w))$ is the index of function $g_j(\cdot)$ presenting the greatest value at w.

Since g_k is convex, for all u

$$g_k(u) \geq g_k(w) + \langle u - w, \nabla g_k(w) \rangle$$

However $f(u) = \max_{\forall j}(g_j(u)) \geq g_k(u)$ for any j, and $f(w) = g_k(w)$ at w. Then

$$\begin{aligned} f(u) \geq & g_k(u) \\ \geq & g_k(w) + \langle u - w, \nabla g_k(w) \rangle \\ = & f(w) + \langle u - w, \nabla g_k(w) \rangle \end{aligned}$$

Then by the definition of the sub-gradient $\nabla g_k(w) \in \partial f(w)$

The following is given as a homework (Formative assessment 1)

Exercise 6. ($\star$)Consider the binary classification problem with inputs $x \in \mathcal{X}$ where $\mathcal{X} := \left\{x \in \mathbb{R}^d : \|x\|_2 \leq L\right\}$ for some given value $L > 0$, target $y \in \mathcal{Y}$ where $\mathcal{Y} := \{-1, +1\}$, and prediction rule $h_w : \mathbb{R}^d \to \{-1, +1\}$ with

$$h_w(x) = \text{sign}\left(w^\top x\right) \tag{2}$$

$$= \text{sign}\left(\sum_{j=1}^{d} w_j x_j\right) \tag{3}$$

Let the hypothesis class of prediction rules be

$$\mathcal{H} = \left\{x \to w^\top x : \forall w \in \mathbb{R}^d\right\}$$

In other words, the hypothesis $h_w \in \mathcal{H}$ is parametrized by $w \in \mathbb{R}^d$ it receives an input vector $x \in \mathcal{X} := \mathbb{R}^d$ and it returns the label $y = \text{sign}\left(w^\top x\right) \in \mathcal{Y} := \{\pm 1\}$.

Consider a loss function $\ell : \mathbb{R}^d \to \mathbb{R}_+$ with

$$\ell(w, z = (x, y)) = \max\left(0, 1 - y w^\top x\right) + \lambda \|w\|_2^2 \tag{4}$$

for some given value $\lambda > 0$.

Assume there is available a dataset of examples $S_n = \{z_i = (x_i, y_i); i = 1, ..., n\}$ of size n.

Do the following tasks.

Hint-1:: We denote

$$\text{sign}(\xi) = \begin{cases} -1, & \text{if } \xi < 0 \\ +1, & \text{if } \xi > 0 \end{cases}$$

Hint-2:: The notation ±1 means either -1 or $+1$.

HInt-3:: We define $\mathbb{R}_+ := (0, +\infty)$

Hint-4:: We denote $\|x\|_2 := \sqrt{\sum_{\forall j}(x_j)^2}$ the Euclidean distance.

(1) Show that the function $f : \mathbb{R} \to \mathbb{R}_+$ with $f(x) = \max(0, 1 - x)$ is convex in $\mathbb{R}$; and show that the loss (4) is convex.

Hint:: You may use Example 13 from Handout 1.

(2) Show that the loss $\ell(w,z)$ for $\lambda=0$ (4) is L-Lipschitz (with respect to w) when $x\in\mathcal{X}$ where $\mathcal{X}:=\left\{x\in\mathbb{R}^d : \|x\|_2\le L\right\}$.

Hint:: You may use the definition of Lipschitz function. Without loss of generality, you can consider any $w_1\in\mathbb{R}^d$ and $w_2\in\mathbb{R}^d$ such that $1-yw_2^\top x\le 1-yw_1^\top x$, and then take cases $1-yw_2^\top x$ $>$ or $<$ 0 and $1-yw_1^\top x$ $>$ or $<$ 0 to deal with the max.

(3) Construct the set of sub-gradients $\partial f(x)$ for $x\in\mathbb{R}$ of the function $f:\mathbb{R}\to\mathbb{R}_+$ with $f(x)=\max(0,1-x)$. Show that the vector v with

$$v=\begin{cases}2\lambda w, & yw^\top x>1\\ 2\lambda w, & yw^\top x=1\\ -yx+2\lambda w, & yw^\top x<1\end{cases}$$

is $v\in\partial_w\ell(w,z=(x,y))$, aka a sub-gradient of $\ell(w,z=(x,y))$ at w, for any $w\in\mathbb{R}^d$.

(4) Write down the algorithm of online AdaGrad (Adaptive Stochastic Gradient Descent) with learning rate $\eta_t>0$, batch size m, and termination criterion $t>T_{\max}$ for some $T_{\max}>0$ in order to discover w^* such as

$$w^*=\arg\min_{\forall w:h_w\in\mathcal{H}}\left(\mathrm{E}_{z\sim g}\left(\ell\left(w,z=(x,y)\right)\right)\right) \tag{5}$$

The formulas in your algorithm have to be tailored to 4.

(5) Use the R code given below in order to generate the dataset of observed examples $S_n=\{z_i=(x_i,y_i)\}_{i=1}^n$ that contains $n=10^6$ examples with inputs x of dimension $d=2$. Consider $\lambda=0$. Use a seed $w^{(0)}=(0,0)^\top$.

(a) By using appropriate values for m, η_t and $T_{\max}$, code in R the algorithm you designed in part 4, and run it.

(b) Plot the trace plots for each of the dimensions of the generated chain $\left\{w^{(t)}\right\}$ against the iteration t.

(c) Report the value of the output w^*_{adaGrad} (any type) of the algorithm as the solution to (5).

(d) To which cluster y (i.e., -1 or 1) $x_{\text{new}}=(1,0)^\top$ belongs?

```
# R code. Run it before you run anything else
#
data_generating_model <- function(n,w) {
z <- rep( NaN, times=n*3 )
z <- matrix(z, nrow = n, ncol = 3)
z[,1] <- rep(1,times=n)
z[,2] <- runif(n, min = -10, max = 10)
p <- w[1]*z[,1] + w[2]*z[,2] p <- exp(p) / (1+exp(p))
z[,3] <- rbinom(n, size = 1, prob = p)
ind <- (z[,3]==0)
z[ind,3] <- -1
x <- z[,1:2]
y <- z[,3]
return(list(z=z, x=x, y=y))
}
n_obs <- 1000000
w_true <- c(-3,4)
set.seed(2023)
out <- data_generating_model(n = n_obs, w = w_true)
set.seed(0)
z_obs <- out$z #z=(x,y)
x <- out$x
y <- out$y
#z_obs2=z_obs
#z_obs2[z_obs[,3]==-1,3]=0
#w_true <- as.numeric(glm(z_obs2[,3]~ 1+ z_obs2[,2],family = "binomial"
)$coefficients)
```

Solution.

(1) $f_1(x) = 0$ is convex, $f_2(x) = 1 - x$ is convex, hence from the example in Handout 1, $f(x) = \max(f_1(x), f_2(x))$ is convex as well. Regarding the loss function, we just have $f_2(w) = 1 - yx^\top w$ which is convex as a composition due to linearity.

(2) Given a fixed example $(x, y) \in \left\{x \in \mathbb{R}^d : \|x'\|_2 \leq R\right\} \times \{-1, 1\}$.

Assume $w_1, w_2 \in \mathbb{R}^d$. Let $\ell_i = \max\left\{0, 1 - yx^\top w_i\right\}$, for $i = 1, 2$. It suffices to show that $|\ell_1 - \ell_2|_2 \leq R\,|w_1 - w_2|_2$. I take cases

Case-1: Assume $yx^\top w_1 \geq 1$ and $yx^\top w_2 \geq 1$ then $|\ell_1 - \ell_2|_2 = 0 \leq R\,|w_1 - w_2|_2$

Case-2: Assume that at least one of $yx^\top w_1 < 1$ or $yx^\top w_2 < 1$ but not both is true. Assume without loss of generality that $1 - yx^\top w_1 < 1 - yx^\top w_2$. Then

$$\begin{aligned}
|\ell_1 - \ell_2|_2 =& \ell_1 - \ell_2 \\
=& 1 - yx^\top w_1 - \max\left(0, 1 - yx^\top w_2\right) \\
\leq& 1 - yx^\top w_1 - \left(1 - yx^\top w_2\right) \\
=& yx^\top \left(w_2 - w_1\right) \\
\leq& y \left\|x^\top\right\|_2 \|w_1 - w_2\|_2 \quad \text{because} \quad a^\top b \leq \|a\| \, \|b\|
\end{aligned}$$

(3) It is

$$f(x) = \max(0, 1 - x) = \begin{cases} 0 & x > 1 \\ 0 & x = 1 \\ 1 - x & x < 1 \end{cases}$$

- For $x > 1$, f is differentiable so $\partial f(x) = \{f'(x)\} = \{0\}$.
- For $x < 1$, f is differentiable so $\partial f(x) = \{f'(x)\} = \{-1\}$.
- For $x = 1$, f is not differentiable. By definition I have that v is subgradient of $f(x)$ at $x = 0 \in S$ if

  $$\forall u \in \mathbb{R}, \quad f(u) \geq f(x) + \langle u - x, v \rangle$$

  So, for $u \geq 1$, it is $0 \geq (u - 1) v \implies v \leq 0$, and for $u < 1$ it is $(1 - u) \geq (u - 1) v \implies v \geq -1$. Hence the common space is $v \in [0, 1]$ So $\partial f(x) = [0, 1]$. Hence,

  $$\partial f(x) = \begin{cases} 0, & x > 1 \\ [-1, 0], & x = 1 \\ -1, & x < 1 \end{cases}$$

Now regarding the loss $\partial_w \ell(w, z = (x, y))$

- for $yw^\top x > 1$ it is differentiable so $\nabla_w \ell(w, z = (x, y)) = \nabla_w \left(0 + \lambda \sum_{j=1}^d w_j^2\right) = 2\lambda w$; as

  $$\frac{\mathrm{d}}{\mathrm{d}w_j} \sum_{j'=1}^d w_{j'}^2 = 2\lambda w_j$$

- for $yw^\top x > 1$ it is differentiable so $\nabla_w \ell(w, z = (x, y)) = \nabla_w \left(1 - yw^\top x + \lambda \sum_{j=1}^d w_j^2\right) = yx + 2\lambda w$ as

  $$\frac{\mathrm{d}}{\mathrm{d}w_j} \left(1 - yw^\top x\right) = \frac{\mathrm{d}}{\mathrm{d}w_j} \left(1 - y \sum_{j'=1}^d w_{j'} x_{j'}\right) = -yx_j$$

- for $yw^\top x = 1$, $v = 0$ satisfies the definition of the sub-gradient

$$\begin{aligned} \forall u,\ \ f\left(u\right) &\geq \cancelto{0}{f\left(w\right)} + \langle u - w, v\rangle \\ \max\left(0, 1 - yu^\top x\right) &\geq 0 + \left(u - w\right)^\top 0 \end{aligned}$$

So

$$\begin{aligned} \partial \ell\left(w, z = (x, y)\right) &= \partial\left(\max\left(0, 1 - yw^\top x\right) + \lambda \left\|w\right\|_2^2\right) \\ &= \partial\left(\max\left(0, 1 - yw^\top x\right)\right) + \partial\left(\lambda \left\|w\right\|_2^2\right) \\ &= \partial\left(\max\left(0, 1 - yw^\top x\right)\right) + \nabla\left(\lambda \left\|w\right\|_2^2\right) \\ & 0 + 2\lambda w \end{aligned}$$

but $\partial\left(\lambda \left\|w\right\|_2^2\right) = \left\{\nabla\left(\lambda \left\|w\right\|_2^2\right)\right\}$ because $\lambda \left\|w\right\|_2^2$ is differentiable. Hence

$$\partial \ell\left(w, z = (x, y)\right) = 0 + 2\lambda w$$

Hence

$$v = \begin{cases} 2\lambda w, & yw^\top x > 1 \\ 2\lambda w, & yw^\top x = 1 \\ -yx + 2\lambda w, & yw^\top x < 1 \end{cases}$$

(4)

Algorithm. *For $t = 1, 2, 3, ...$ iterate:*

(a) *Get a random sub-sample $\left\{\tilde{z}_i^{(t)} = \left(\tilde{x}_i^{(t)}, \tilde{y}_i^{(t)}\right); i = 1, ..., m\right\}$ of size m with or without replacement from the complete data-set $\mathcal{S}_n$.*

(b) *For $j = 1, ..., d$ (index j indicates the dimension of w) compute*

$$w_j^{(t+1)} = w_j^{(t)} - \eta_t \frac{1}{\sqrt{\left[G_t\right]_{j,j} + \epsilon}} \bar{v}_{t,j}$$

$\left[G_t\right]_{j,j} = \left[G_{t-1}\right]_{j,j} + \left(\bar{v}_{t,j}\right)^2$ *where* $\bar{v}_t = \frac{1}{m}\sum_{i=1}^m \tilde{v}_{t,i}$ *and*

$$\tilde{v}_{t,i} = \begin{cases} 2\lambda w^{(t)}, & \tilde{y}_i^{(t)}\left(w^{(t)}\right)^\top \tilde{x}_i^{(t)} > 1 \\ 2\lambda w^{(t)}, & \tilde{y}_i^{(t)}\left(w^{(t)}\right)^\top \tilde{x}_i^{(t)} = 1 \\ -\frac{1}{m}\tilde{y}_i^{(t)}\tilde{x}_i^{(t)} + 2\lambda w^{(t)}, & \tilde{y}_i^{(t)}\left(w^{(t)}\right)^\top \tilde{x}_i^{(t)} < 1 \end{cases}$$

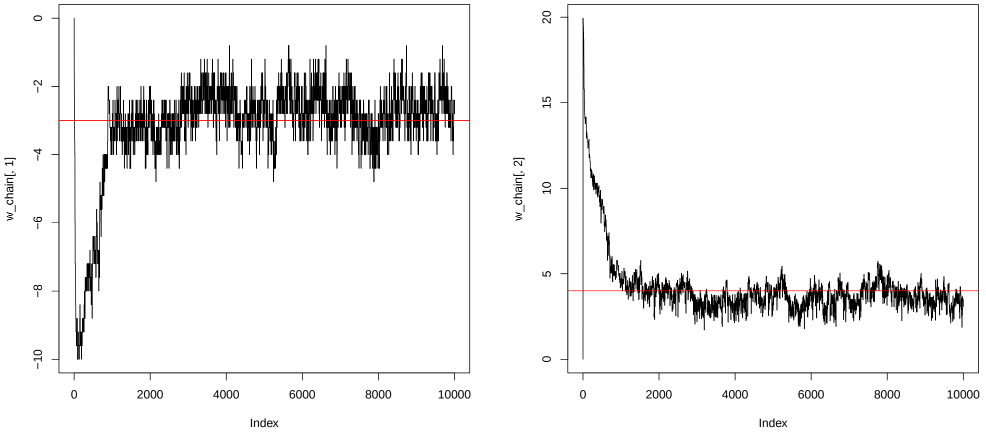
where index i indicates the sub-sample, and $\epsilon > 0$ small.

(c) *Terminate if a termination criterion is satisfied*

(5)

(a) The R code can be found in the link `https://raw.githubusercontent.com/georgios-stats/Machine_Learning_and_Neural_Networks_III_Epiphany_2023/main/Exercises/supplementary/q6.R`

(b) The figures are presented below



(A) $w_1^{(t)}$ (B) $w_2^{(t)}$

Figure 1. trace plots

(c) I found $w = (-2.674615, 3.205785)$

(d) It belongs to -1

Exercise 7. ($\star$)Assume a Bayesian model

$$\begin{cases} z_i|w & \overset{\text{ind}}{\sim} f\left(z_i|w\right),\ i = 1,...,n \\ w & \sim f\left(w\right) \end{cases}$$

and consider that our objective is the discovery of MAP estimate w^* i.e.

$$w^* = \arg\min_{\forall w\in\Theta}\left(-\log\left(L_n\left(w\right)\right) - f\left(w\right)\right) = \arg\min_{\forall w\in\Theta}\left(-\sum_{i=1}^{n}\log\left(f\left(z_i|w\right)\right) - \log\left(f\left(w\right)\right)\right)$$

by using SGD with update

$$w^{(t+1)} = w^{(t)} + \eta_t\left(\frac{n}{m}\sum_{j\in\mathcal{J}^{(t)}}\nabla_w\log\left(f\left(z_j|w^{(t)}\right)\right) + \nabla_w\log\left(f\left(w^{(t)}\right)\right)\right)$$

for some randomly selected set $\mathcal{J}^{(t)} \subseteq \{1,...,n\}^m$ of m integers from 1 to n via simple random sampling (SRS) with replacement. Show that

$$\mathrm{E}_{\mathcal{J}^{(t)}\sim\text{simple-random-sampling}}\left(\frac{n}{m}\sum_{j\in\mathcal{J}^{(t)}}\nabla_w\log\left(f\left(z_j|w^{(t)}\right)\right)\right) = \sum_{i=1}^{n}\nabla_w\log\left(f\left(z_i|w^{(t)}\right)\right)$$

Solution. It is

$$\begin{aligned}
\mathrm{E}_{\mathcal{J}^{(t)}\sim\mathrm{SRS}}\left(\frac{n}{m}\sum_{j\in\mathcal{J}^{(t)}}\nabla_w\log\left(f\left(z_j|w^{(t)}\right)\right)\right) =& \frac{n}{m}\sum_{j\in\mathcal{J}^{(t)}}\mathrm{E}_{\mathcal{J}^{(t)}\sim\mathrm{SRS}}\left(\nabla_w\log\left(f\left(z_j|w^{(t)}\right)\right)\right)\\
=& \frac{n}{m}\sum_{j\in\mathcal{J}^{(t)}}\mathrm{E}_{\mathcal{J}^{(t)}\sim\mathrm{SRS}}\left(\nabla_w\log\left(f\left(z_j|w^{(t)}\right)\right)\right)\\
=& \frac{n}{m}\sum_{j\in\mathcal{J}^{(t)}}\frac{1}{n}\sum_{i=1}^{n}\nabla_w\log\left(f\left(z_i|w^{(t)}\right)\right)\\
=& \sum_{i=1}^{n}\nabla_w\log\left(f\left(z_i|w^{(t)}\right)\right)
\end{aligned}$$

It is $\mathrm{E}_{\mathcal{J}^{(t)}\sim\mathrm{SRS}}\left(\nabla_w\log\left(f\left(z_j|w^{(t)}\right)\right)\right) = \frac{1}{n}\sum_{i=1}^{n}\nabla_w\log\left(f\left(z_i|w^{(t)}\right)\right)$ because the expectation is under the probability I get randomly an integer and for the jth on the probability is $1/n$ due to the random scheme. Also $\left|\mathcal{J}^{(t)}\right| = m$.

Part 2. **Artificial Neural Networks**

Exercise 8. ($\star$) Students are encouraged to practice on the Exercises 5.1-5.28 from the textbook

- Bishop, C. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: Springer.

Exercise 9. available from

- `https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-`
  `pdf`
- https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf

The solutions are available from

- `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/05/prml-web-sol-2009-09-`
  `pdf`
- https://www.microsoft.com/en-us/research/wp-content/uploads/2016/05/prml-web-sol-2009-09-08.pdf

---

The following is given as a homework (Formative assessment 2)

Exercise 10. ($\star$)Consider the multi-class classification problem, with a predictive rule $h_w : \mathbb{R}^d \to \mathcal{P}$, as a classification probability i.e, $h_{w,k}(x) = \Pr(x \text{ belongs to class } k)$, that receives values $x \in \mathbb{R}^d$ returns vales in $\mathcal{P} = \left\{p \in (0,1)^q : \sum_{j=1}^{q} p_j = 1\right\}$. We assume $h_w = (h_{w,1}, ..., h_{w,q})^\top$, and modeled as an ANN

$$h_k\left(x\right)=\sigma_2\left(\sum_{j=1}^{c}w_{2,k,j}\sigma_1\left(\sum_{i=1}^{d}w_{1,j,i}x_i\right)\right)$$

for $k=1,...,q$, with activation functions softmax function

$$\sigma_2\left(a_k\right)=\frac{\exp\left(a_k\right)}{\sum_{k'=1}^{q}\exp\left(a_{k'}\right)},\text{ for } k=1,...,q$$

and $\sigma_1\left(a\right)=\arctan\left(a\right)$. Consider a loss

$$\ell\left(w,z=\left(x,y\right)\right)=-\sum_{k=1}^{q}y_k\log\left(h_{w,k}\left(x\right)\right)$$

at w and example $z=\left(x,y\right)$, where $x\in\mathbb{R}^d$ is the input vector (features), and $y=\left(y_1,...,y_q\right)$ is the output vector (labels) with $y\in\{0,1\}^q$ and $\sum_{k=1}^{q}y_k=1$. Consider that d, c, and q are known quantities.

Hint: You may use

$$\frac{\mathrm{d}}{\mathrm{d}x}\arctan\left(x\right)=\frac{1}{1+x^2}$$

(1) Perform the forward pass of the back-propagation procedure to compute the activations which may be denoted as $\{a_{t,i}\}$ and outputs which may be denoted as $\{o_{t,i}\}$ at each layer t.

(2) Show that

$$\frac{\mathrm{d}}{\mathrm{d}a_k}\sigma_2\left(a_j\right)=\sigma_2\left(a_j\right)\left(1\left(j=k\right)-\sigma_2\left(a_k\right)\right)$$

for $k=1,...,q$. Let $1\left(j=k\right)=\begin{cases}1 & j=k\\ 0 & j\neq k\end{cases}$.

(3) Perform the backward pass of the back-propagation procedure in order to compute the elements of the gradient $\nabla_w\ell\left(w,\left(x,y\right)\right)$.

Solution. Forward pass

```
Set: $o_{0,i}=x_i$ for $i=1,...,d$
Compute:
    at $t=1$: for $j=1,...c$
        comp: $\alpha_{1,j}=\sum_{i=1}^{d}w_{1,i,j}x_i$
        comp: $o_{1,j}=\arctan\left(\alpha_{1,j}\right)$
    at $t=2$: for $k=1,...q$
        comp: $\alpha_{2,k}=\sum_{j=1}^{d}w_{2,k,j}o_{2,j}$
        comp: $o_{2,k}=\frac{\exp\left(\alpha_{2,k}\right)}{\sum_{k'=1}^{q}\exp\left(\alpha_{2,k}\right)}$
    get: $h_k=o_{2,k}$
```

(1) It is

$$\frac{\mathrm{d}}{\mathrm{d}a_k}\sigma_2\left(a_j\right)=\frac{\mathrm{d}}{\mathrm{d}a_k}\frac{\exp\left(a_j\right)}{\sum_{j'}\exp\left(a_{j'}\right)}=\begin{cases}\sigma_2\left(a_j\right)\left(1-\sigma_2\left(a_j\right)\right) & j=k\\ -\sigma_2\left(a_j\right)\sigma_2\left(a_k\right) & j\neq k\end{cases}$$
$$=\sigma_2\left(a_j\right)\left(1\left(j=k\right)-\sigma_2\left(a_k\right)\right)$$

(2) It is

$$\frac{\mathrm{d}}{\mathrm{d}a}\sigma_1\left(a\right)=\frac{1}{1+a^2}$$

and

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}a_k}\sigma_2\left(a_k\right)&=\sigma_2\left(a_j\right)\left(1\left(j=k\right)-\sigma_2\left(a_k\right)\right)\\&=o_j\left(1\left(j=k\right)-o_k\right)\end{aligned}$$

and

$$\frac{\mathrm{d}\ell_2}{\mathrm{d}o_{2,j}}=-y_j\frac{1}{o_{2,j}}$$

and

$$\begin{aligned}\frac{\mathrm{d}\ell_2}{\mathrm{d}a_{2,k}}&=\sum_{j=1}^{q}\frac{\mathrm{d}\ell_2}{\mathrm{d}o_{2,j}}\frac{\mathrm{d}o_{2,j}}{\mathrm{d}o_{2,k}}\\&=\sum_{j=1}^{q}\left(-y_j\frac{1}{o_{2,j}}o_{2,j}\left(1\left(j=k\right)-o_{2,k}\right)\right)\\&=\sum_{j=1}^{q}\left(-y_j\left(1\left(j=k\right)-o_{2,k}\right)\right)\\&=o_{2,k}-y_k\end{aligned}$$

Backward pass:

at $t=2$**:** for $k=1,...q$

comp: $\tilde{\delta}_{2,k}=\frac{\mathrm{d}}{\mathrm{d}\alpha_{2,k}}\ell_T=o_{2,k}-y_k$

at $t=1$**:** for $j=1,...c$

comp:

$$\begin{aligned}\tilde{\delta}_{1,j}&=\left.\frac{\mathrm{d}}{\mathrm{d}\xi}\sigma_1\left(\xi\right)\right|_{\xi=\alpha_{1,j}}\sum_{k=1}^{q}w_{2,k,j}\tilde{\delta}_{2,k}\\&=\left(\frac{1}{1+\alpha_{1,j}^2}\right)\sum_{k=1}^{q}w_{2,k,j}\tilde{\delta}_{2,k}\end{aligned}$$

Output:

$$\frac{\mathrm{d}}{\mathrm{d}w_{1,j,i}}\ell=\tilde{\delta}_{1,j}x_i\text{ and }\frac{\mathrm{d}}{\mathrm{d}w_{2,k,j}}\ell=\tilde{\delta}_{2,k}o_{2,j}$$