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.

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.

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 $\langle y,x\rangle\leq\left\Vert y\right\Vert \left\Vert x\right\Vert$

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

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

Exercise 5. ($\star$)Let $g_{1}\left(w\right),...,g_{r}\left(w\right)$ be r convex functions, and let $f\left(\cdot\right)=\max_{\forall j}\left(g_{j}\left(\cdot\right)\right)$. Show that for some w it is $\nabla g_{k}\left(w\right)\in\partial f\left(w\right)$ where $k=\arg\max_{j}\left(g_{j}\left(w\right)\right)$ is the index of function $g_{j}\left(\cdot\right)$ presenting the greatest value at 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 $\mathcal{X}:=\left\{ x\in\mathbb{R}^{d}:\left\Vert x\right\Vert _{2}\leq L\right\}$ for some given value $L>0$, target $y\in\mathcal{Y}$ where $\mathcal{Y}:=\{-1,+1\}$, and prediction rule $h_{w}:\mathbb{R}^{d}\rightarrow\{-1,+1\}$ with

$$h_{w}\left(x\right)=\text{sign}\left(w^{\top}x\right) \tag{1}$$

$$=\text{sign}\left(\sum_{j=1}^{d}w_{j}x_{j}\right) \tag{2}$$

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

$$\ell\left(w, z = (x, y)\right) = \max\left(0, 1 - y w^{\top} x\right) + \lambda \|w\|_2^2 \tag{3}$$

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

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

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

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

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

(1) Show that the function $f : \mathbb{R} \to \mathbb{R}_+$ with $f\left(x\right) = \max\left(0, 1 - x\right)$ is convex in $\mathbb{R}$; and show that the loss (3) is convex.

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

(2) Show that the loss $\ell\left(w, z\right)$ for $\lambda = 0$ (3) is L-Lipschitz (with respect to w) when $x \in \mathcal{X}$ where $\mathcal{X} := \left\{x \in \mathbb{R}^d : \|x\|_2 \leq L\right\}$.

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 $1 - y w_2^{\top} x \leq 1 - y w_1^{\top} x$, and then take cases $1 - y w_2^{\top} x\ >$ or $<\ 0$ and $1 - y w_1^{\top} x >$ or < 0 to deal with the max.

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

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

is $v \in \partial_w \ell\left(w, z = (x, y)\right)$, aka a sub-gradient of $\ell\left(w, z = (x, y)\right)$ 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

$$w^* = \arg \min_{\forall w : h_w \in \mathcal{H}} \left(\mathrm{E}_{z \sim g}\left(\ell\left(w, z = (x, y)\right)\right)\right) \tag{4}$$

The formulas in your algorithm have to be tailored to 3.

(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 $\{w^{(t)}\}$ against the iteration t.

(c) Report the value of the output w^*_{adaGrad} (any type) of the algorithm as the solution to (4).

(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)
```

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

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

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

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

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

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

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.

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
- `https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-1.pdf`
  - https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20ma
    solutions-1.pdf
- `https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-2.pdf`

– https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20ma
solutions-2.pdf

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

Exercise 9. ($\star$)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

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

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

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

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

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

at w and example $z = (x, y)$, where $x \in \mathbb{R}^d$ is the input vector (features), and $y = (y_1, ..., y_q)$ 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

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

(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

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

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

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

Part 3. Support Vector Machines

The following is given as a homework (Formative assessment 3)

Exercise 10. ($\star\star$) Consider a training data set $\mathcal{D} = \{z_i = (x_i, y_i)\}_{i=1}^m$. Consider the Soft-SVM Algorithm that requires the solution of the following quadratic minimization problem (in a slightly modified but equivalent form to what we have discussed)

Primal problem:

$$(w^*, b^*, \xi^*) = \underset{(w,b,\xi)}{\arg\min} \left(\frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \right) \tag{5}$$

$$\text{subject to: } y_i \left(\langle w, x_i \rangle + b \right) \geq 1 - \xi_i, \;\; \forall i = 1, ..., m \tag{6}$$

$$\xi_i \geq 0, \;\; \forall i = 1, ..., m \tag{7}$$

for some user-specified fixed parameter $C > 0$. We seek to find the dual problem of 5-7.

(1) Specify the Lagrangian function L associated to the above primal quadratic minimization problem, where $\{\alpha_i\}$ are the Lagrange coefficients wrt (6), and $\{\beta_i\}$ are the Lagrange coefficients wrt (7). Write down any possible restrictions on the Lagrange coefficients.
(2) Compute the dual Lagrangian function denoted as $\tilde{L}$ as a function of the Lagrange coefficients and the data points $\mathcal{D}$.
(3) Apply the KKT conditions to the above problem, and write them down.
(4) Derive and write down the dual Lagrangian quadratic maximization problem, along with the inequality and equality constraints, where you seek to find $\{\alpha_i\}$.
(5) Justify why the i-th point x_i lies on the margin boundary when $\alpha_i \in (0, C)$ (beware it is $\alpha_i \neq C$), and why the i-th point x_i lies inside the margin when $\alpha_i = C$.
(6) Given optimal values $\{\alpha_i^*\}$ for Lagrangian coefficients $\{\alpha_i\}$ as they are derived by solving the dual Lagrangian maximization problem in part 4, derive the optimal values w^* and b^* for the parameters w and b as function of the support vectors. Regarding parameter b it should be in the derived in the form
$$b^* = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \left(y_i - \sum_{j \in \mathcal{S}} \alpha_j^* y_j \langle x_j, x_i \rangle \right)$$
where you determine the sets $\mathcal{M}$ and $\mathcal{S}$.
(7) Report the halfspace predictive rule $h_{w,b}(x)$ of the above problem as a function of α^* and b^*.

Exercise 11. ($\star$) Students are encouraged to practice on the Exercises 6.1-6.19 from the textbook

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

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
- `https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%`
  `20and%20machine%20learning-solutions-1.pdf`
  - https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20ma
    solutions-1.pdf
- `https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%`
  `20and%20machine%20learning-solutions-2.pdf`
  - https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20ma
    solutions-2.pdf

Part 4. **Gaussian process regression**

Exercise 12. ($\star$) Students are encouraged to practice on the Exercises 6.19-6.27 from the textbook

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

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
- `https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%`
  `20and%20machine%20learning-solutions-1.pdf`
  - https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20ma
    solutions-1.pdf
- `https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%`
  `20and%20machine%20learning-solutions-2.pdf`
  - https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20ma
    solutions-2.pdf
