

## Revision sheet

Lecturer &amp; Author: Georgios P. Karagiannis

georgios.karagiannis@durham.ac.uk

**Exercise 1.** Consider a prediction rule  $h : \mathbb{R}^d \rightarrow \mathbb{R}_+^q$  with  $h(x) = (h_1(x), \dots, h_q(x))^\top$  which receives inputs  $x = (x_1, \dots, x_d)^\top \in \mathbb{R}^d$  and which is modeled as a feedforward neural network (NN) with equation

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

for  $k = 1, \dots, q$ . We consider activation functions  $\sigma_1(a) = \frac{1}{1+\exp(-a)}$  and  $\sigma_2(a) = \log(1 + \exp(a))$ . Parameters  $c \in \mathbb{N}_+$ , and  $d \in \mathbb{N}_+$  are considered as known, while the weights  $\{w_{\cdot,\cdot,\cdot}\}$  of the NN are unknown. To learn the unknown weights  $\{w_{\cdot,\cdot,\cdot}\}$ , we specify the loss function

$$\ell(w, z = (x, y)) = \frac{1}{2} \|h(x) - y\|_2^2 = \frac{1}{2} \sum_{k=1}^q (h_k(x) - y_k)^2$$

where  $z = (x, y)$  denotes an example,  $x \in \mathbb{R}^d$  is the input vector (features), and  $y = (y_1, \dots, y_q)^\top \in \mathbb{R}^q$  is the output vector (targets).

- (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) Perform the backward pass of the back-propagation procedure in order to compute the gradient

$$\nabla_w \ell(w, (x, y)) = \left( \left( \frac{\partial}{\partial w_{1,j,i}} \ell(w, (x, y)) \right)_{j=1, i=1}^{c,d}, \left( \frac{\partial}{\partial w_{2,k,j}} \ell(w, (x, y)) \right)_{k=1, j=1}^{q,c} \right)$$

of the loss function  $\ell(w, z)$  with respect to  $w$  for any example  $z = (x, y)$ . Clearly state the steps of the procedure as well as state the quantities

$$\frac{\partial}{\partial w_{1,j,i}} \ell(w, (x, y)), \text{ and } \frac{\partial}{\partial w_{2,k,j}} \ell(w, (x, y))$$

for all  $k = 1, \dots, q$ ,  $j = 1, \dots, c$ , and  $i = 1, \dots, d$ .

**Solution.** I've got  $T = 2$  layers.

- (1) Regarding the forward pass. It is

**Set:** Comp for  $i = 1, \dots, d$

$$o_{0,i}(x) = x_i$$

**t=1:** Comp

$$\alpha_{1,j}(x) = \sum_{i=1}^d w_{1,j,i} x_i$$

$$o_{1,j}(x) = (1 + \exp(-\alpha_{1,j}(x)))^{-1}$$

**t=2:** Comp

$$\alpha_{2,k}(x) = \sum_{j=1}^c w_{2,k,j} o_{1,j}(x)$$

$$o_{2,k}(x) = \log(1 + \exp(\alpha_{2,k}(x)))$$

**Get:** Comp for  $k = 1, \dots, q$

$$h_k(x) = o_{2,k}(x)$$

(2) Regarding the backward pass. It is

$$\frac{d}{d\xi} \sigma_2(\xi) = \frac{\exp(\xi)}{1 + \exp(\xi)} = \sigma_1(\xi)$$

and

$$\begin{aligned} \frac{d}{d\xi} \sigma_1(\xi) &= -\frac{\exp(-\xi)}{1 + \exp(-\xi)} \frac{1}{1 + \exp(-\xi)} \\ &= -\sigma_1(\xi) (1 - \sigma_1(\xi)) \end{aligned}$$

**t=T=2:**

$$\begin{aligned} \tilde{\delta}_{T=2,k} &= \frac{d}{do_{T,k}} \ell_T(w, z) \frac{do_{T,k}}{d\alpha_{T,k}} \\ &= (o_{T,k} - y_k) \sigma_1(\alpha_{T,k}) \\ &= (h_k - y_k) \sigma_1(\alpha_{T,k}) \end{aligned}$$

or for  $k = 1, \dots, q$

$$\tilde{\delta}_{2,k} = (h_k - y_k) \sigma_1(\alpha_{T,k})$$

**t=1:**

$$\begin{aligned} \tilde{\delta}_{t=1,k} &= \sum_{j=1}^q w_{1,k,j} \tilde{\delta}_{2,k} \left. \frac{d}{d\xi} \sigma_1(\xi) \right|_{\xi=\alpha_j} \\ &= -\sigma_1(\alpha_j) (1 - \sigma_1(\alpha_j)) \left[ \sum_{k=1}^q w_{1,k,j} \tilde{\delta}_{2,k} \right] \end{aligned}$$

Hence

$$\begin{aligned} \frac{d}{dw_{2,k,j}} \ell(w, (x, y)) &= (h_k - y_k) \sigma_1(\alpha_{2,k}) o_{1,j}(x) \\ \frac{d}{dw_{1,j,i}} \ell(w, (x, y)) &= -\sigma_1(\alpha_j) (1 - \sigma_1(\alpha_j)) \left[ \sum_{k=1}^q w_{1,k,j} \tilde{\delta}_{2,k} \right] o_{0,i}(x) \end{aligned}$$

**Exercise 2.** Consider the binary classification learning problem: Let the set of targets be  $\mathcal{Y} = \{-1, +1\}$ , let the set of inputs be  $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \leq B\}$  for some scalar  $B > 0$ , let the prediction rule be  $h_w(x) = x^\top w$ , and let the loss function  $\ell$  be

$$\ell(w, z = (x, y)) = \log \left( 1 + \exp \left( -yx^\top w \right) \right),$$

for  $x \in \mathcal{X}$ ,  $y \in \mathcal{Y}$ , and  $w \in \mathcal{W}$  where  $\mathcal{W} = \{w \in \mathbb{R}^d : \|w\|_2 \leq B\}$ .

- (1) Show that the resulting learning problem is convex-Lipschitz-bounded. Specify the parameter of Lipschitzness.
- (2) Show that the above loss  $\ell(w, z = (x, y))$  is  $B^2/4$  smooth.

**Hint::** You may use the mean value theorem which states that (under pre-assumed conditions),  $f(b) - f(a) = \frac{d}{dx} f(x)|_{x=c} (b - a)$  for  $a \leq c \leq b$ .

- (3) Consider a risk function  $R_g(w) = \mathbb{E}_{z \sim g} (\ell(w, z = (x, y)))$  where  $g$  denotes the unknown data generating process. Assume there is a set of available examples  $\mathcal{D} = \{z_i = (x_i, y_i) ; i = 1, \dots, n\}$ . Now assume that  $w \in \mathbb{R}^d$ . To learn  $w$ , we aim to compute  $w^* \in \mathbb{R}^d$  such that

$$w^* = \arg \min_w (f(w))$$

where  $f(w) = R_g(w) + \frac{\lambda}{2} \|w\|_2^2$ .

- (a) Show that the stochastic gradient descent algorithm with batch size one and with learning rate

$$\eta_t = \frac{1}{\lambda t}$$

at iteration  $t \in \mathbb{N}_+$  which is used to address the learning problem under consideration has a recursion that can be written in the form

$$w^{(t+1)} = -\frac{1}{\lambda t} \sum_{j=1}^t v_j$$

where  $\{v_j\}$  is the gradient of the loss function at certain values of  $w$  and example. Show your working.

- (b) Compute the exact formula of  $v_j$  as a function of  $\lambda$ ,  $t$ ,  $\mathcal{D}$ , and  $\{w^{(t)}\}$ . Show your working.

**Solution.**

(1)

**Convexity::** Note that the function  $g : \mathbb{R} \rightarrow \mathbb{R}$ , defined by  $g(a) = \log(1 + \exp(a))$  is convex. To see this, note that

$$\frac{d^2}{da^2} g(a) = \frac{\exp(a)}{(1 + \exp(a))^2} \geq 0$$

is non-negative. The convexity of  $\ell(\cdot, z)$  for all  $z$  follows as a composition of  $g$  with a linear function.

**Lipschitzness::** The function  $g(a) = \log(1 + \exp(a))$  is 1-Lipschitz since

$$\left| \frac{d}{da} g(a) \right| = \frac{\exp(a)}{1 + \exp(a)} = \frac{1}{\exp(-a) + 1} \leq 1$$

Hence because  $|x|_2 \leq B$ ,  $g(a)$  is  $B$ -Lipschitz as a composition.

**Boundness::** The norm of each hypothesis  $w$  is bounded by  $B$  according to the assumptions.

(2)

**Smoothness::** It is

$$\begin{aligned} \frac{d^2}{da^2} g(a) &= \frac{\exp(a)}{(1 + \exp(a))^2} \\ &= \frac{1}{\exp(a) (1 + \exp(-a))^2} \\ &= \frac{1}{2 + \exp(a) + \exp(-a)} \leq 1/4 \end{aligned}$$

Combine this with the mean value theorem, to conclude that  $\frac{d}{da} g(a)$  is 1/4-Lipschitz. Hence,  $\ell(\cdot, z)$  for all  $z$  is  $B^2/4$  smooth.

(3)

(a) I need to minimize

$$\begin{aligned} f(w) &= R_g(w) + \frac{\lambda}{2} \|w\|_2^2 \\ &= \mathbb{E}_{z \sim g} \left( \ell(w, z = (x, y)) + \frac{\lambda}{2} \|w\|_2^2 \right) \end{aligned}$$

so the online SGD has a recursion

$$w^{(t+1)} = w^{(t)} - \frac{1}{\lambda t} \left( \lambda w^{(t)} + v_t \right)$$

where

$$v_t = \frac{d}{d\xi} \ell \left( \xi, z^{(t)} = (x^{(t)}, y^{(t)}) \right) \Big|_{\xi=w^{(t)}}$$

and  $(x^{(t)}, y^{(t)})$  is a randomly drawn example from the dataset. The recursion is

$$\begin{aligned}
w^{(t+1)} &= w^{(t)} - \frac{1}{\lambda t} (\lambda w^{(t)} + v_t) \\
&= \left(1 - \frac{1}{t}\right) w^{(t)} - \frac{1}{\lambda t} v_t \\
&= \underbrace{\frac{t-1}{t} w^{(t)} - \frac{1}{\lambda t} v_t}_{=\zeta_t} \\
&= \frac{t-1}{t} \left( \frac{t-2}{t-1} w^{(t-1)} - \frac{1}{\lambda(t-1)} v_{t-1} \right) - \frac{1}{\lambda t} v_t \\
&= \frac{t-2}{t-1} w^{(t-1)} - \frac{1}{\lambda t} (v_{t-1} + v_t) \\
&= -\frac{1}{\lambda t} \sum_{j=1}^t v_j
\end{aligned}$$

(b) Regarding the exact formula of  $v_t$

$$v_t = -\frac{\exp\left(-y^{(t)} (x^{(t)})^\top w^{(t)}\right)}{1 + \exp\left(-y^{(t)} (x^{(t)})^\top w^{(t)}\right)} y^{(t)} x^{(t)}$$

where  $(x^{(t)}, y^{(t)})$  is a randomly drawn example from the dataset. This is because

$$\begin{aligned}
\frac{d}{dw} \ell(w, z = (x, y)) &= \frac{d}{dw} \log\left(1 + \exp\left(-yx^\top w\right)\right) \\
&= -\frac{\exp\left(-yx^\top w\right)}{1 + \exp\left(-yx^\top w\right)} yx
\end{aligned}$$