Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Segundo Semestre 2024
Fundamentos Matemáticos para Ciencia de Datos IMT3120

Tarea 3

Instrucciones. Puede usar las conclusiones de una pregunta para responder las preguntas que siguen. Las preguntas 1 y 2 deben entregarse el 20 de Octubre. Las preguntas 3 y 4 deben entregarse el 3 de Noviembre.

Pregunta 1. *(Complejidad de Rademacher de clases afines)*

(a) Considere la clase de funciones lineales-afines acotadas en norma ℓ_2

$$\mathcal{L}_2 = \{h : \mathbb{R}^d \mapsto \mathbb{R} : \ h(x) = w^\top x + b, \ \|w\|_2 \leq R_2, \ |b| \leq r\}$$

Pruebe que si $S = \{X_1, \ldots, X_n\}$ es tal que $S \subseteq \mathcal{B}_2^d(0, L_2)$ entonces

$$\mathcal{R}(\mathcal{L}_2, S) \leq \frac{L_2 R_2}{\sqrt{n}} + \frac{r}{\sqrt{n}}.$$

(b) Considere la clase de funciones lineales-afines acotadas en ℓ_1

$$\mathcal{L}_1 = \{h : \mathbb{R}^d \mapsto \mathbb{R} : \ h(x) = w^\top x + b, \ \|w\|_1 \leq R_1, \ |b| \leq r\}$$

Pruebe que si $S = \{X_1, \ldots, X_n\}$ es tal que $S \subseteq \mathcal{B}_\infty^d(0, L_\infty)$ entonces

$$\mathcal{R}(\mathcal{L}_1, S) \leq L_\infty R_1 \sqrt{\frac{2\ln(2d)}{n}} + \frac{r}{\sqrt{n}}.$$

Pregunta 2. *(Complejidad de Rademacher de redes neuronales)* Sean $\sigma^{(1)}, \ldots, \sigma^{(p)} : \mathbb{R} \mapsto \mathbb{R}$ funciones L-Lipschitz dadas, que llamaremos funciones de activación. Una hipótesis $h : \mathbb{R}^d \mapsto \mathbb{R}$ es expresable por una red neuronal de p capas ocultas[*] si existen $d_0 = d, d_1, \ldots, d_p, d_{p+1} = 1 \in \mathbb{N}$, matrices $W^{(k)} \in \mathbb{R}^{d_k \times d_{k-1}}$ y vectores $b^{(k)} \in \mathbb{R}^{d_k}$ tales que

$$h(x) = h^{(p+1)} \circ \ldots \circ h^{(1)}(x), \tag{1}$$

donde

$$h^{(k)}(y) := \sigma^{(k)}(W^{(k)} y + b^{(k)}) \qquad (\forall k = 1, \ldots, p+1), \tag{2}$$

y donde en la ecuación anterior, la función de activación $\sigma^{(k)}$ es aplicada a cada componente del vector $W^{(k)} y + b^{(k)}$, y $\sigma^{(p+1)}(x) = x$ (i.e., la última capa es lineal-afín).

Denotamos a la clase de hipótesis de funciones expresables por una red neuronal de p capas ocultas y acotadas por $R, r > 0$ como

$$\mathcal{H}_{\text{NN}}^{(p)} := \{h : \mathbb{R}^d \mapsto \mathbb{R} \mid \ h \text{ satisface (1) y (2) con } \|W^{(k)}\|_\infty \leq R \text{ y } \|b^{(k)}\|_\infty \leq r\},$$

donde $\|W\|_\infty = \max_i \sum_j |w_{ij}|$.

[*] Se dice equivalentemente que la red tiene profundidad $p + 1$.

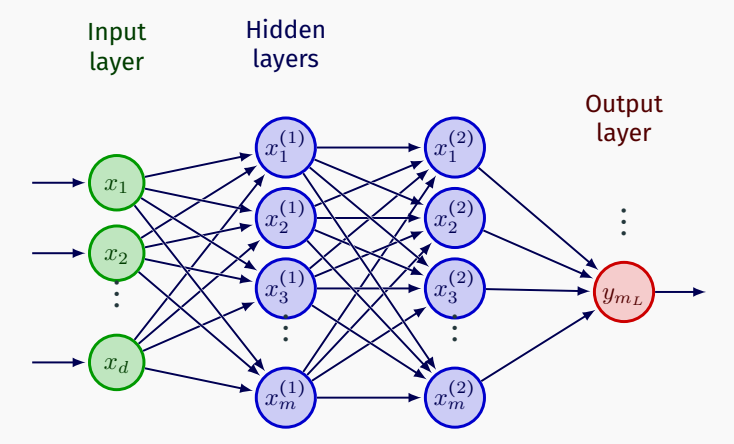

Figura 1: Representación gráfica de una red neuronal con 2 capas ocultas

(a) Sea $\sigma : \mathbb{R} \mapsto \mathbb{R}$ una función de activación L-Lipschitz y $\mathcal{L}$ una clase de hipótesis $\mathbb{R}^d \mapsto \mathbb{R}$ con $0 \in \mathcal{L}$. Definimos la clase

$$\mathcal{H} := \{h : \mathbb{R}^d \mapsto \mathbb{R} \mid h(x) = \sigma\Big(\sum_{j=1}^{m} w_j \ell_j(x) + b\Big),\ \|w\|_1 \le R_1,\ |b| \le r,\ \ell_1, \dots, \ell_m \in \mathcal{L}\}.$$

Pruebe que

$$\mathcal{R}(\mathcal{H}, S) \le L_1\Big[\frac{r}{\sqrt{n}} + 2R_1 \cdot \mathcal{R}(\mathcal{L}, S)\Big].$$

Indicación. Use el álgebra de complejidad de Rademacher visto en ayudantía, además del Lema de Contracción de Talagrand.

(b) Pruebe, usando inducción en p, que

$$\mathcal{R}(\mathcal{H}_{\text{NN}}^{(p)}, S) \le \Big[\frac{r}{\sqrt{n}} + 2RL \cdot \mathcal{R}(\mathcal{H}_{\text{NN}}^{(p-1)}, S)\Big] \quad (\forall p \ge 1),$$

y concluya que

$$\mathcal{R}(\mathcal{H}_{\text{NN}}^{(p)}, S) \le \frac{1}{\sqrt{n}}\Big(r\sum_{k=0}^{p}(2RL)^k + 2R(2RL)^p \max_i \|x_i\|_\infty \sqrt{2\log(2d)}\Big).$$

Pregunta 3. (a) Calcule la función generadora de momentos, $\mathbb{E}[\exp(\lambda \mathbf{x})]$, para $\mathbf{x} \sim \text{Unif}([a, b])$.

(b) Considere el siguiente proceso recursivo $(\mathbf{x}_k)_{k\ge0}$: $\mathbf{x}_0 = 0$ c.s., y para cada $k \ge 0$, $\mathbf{x}_{k+1} \sim \text{Unif}([\mathbf{x}_k, 1])$. Pruebe que $(\mathbf{y}_k)_{k\ge0}$ definido por $\mathbf{y}_k = 2^k(1 - \mathbf{x}_k)$ es una martingala con respecto a $(\mathbf{x}_k)_{k\ge0}$.

(c) Calcule $\mathbb{E}[\exp(\lambda(\mathbf{y}_{k+1} - \mathbf{y}_k))|\mathbf{x}_1, \dots, \mathbf{x}_k]$, es decir, la función generadora de momentos condicional para la diferencia de martingalas $\mathbf{d}_{k+1} := \mathbf{y}_{k+1} - \mathbf{y}_k$.

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Segundo Semestre 2024
Fundamentos Matemáticos para Ciencia de Datos IMT3120

Tarea Parte Computacional

El propósito de esta parte es implementar un método de gradiente estocástico para aprendizaje con redes neuronales.

A continuación describimos algunas características del modelo a utilizar:

(i) Consideramos un modelo de red neuronal con una capa oculta de ancho p_1, donde la función de activación de la capa oculta es la parte positiva (también conocida como ReLU):

$$\sigma(a) = [a]_+ = \text{máx}\{0, a\}.$$

(ii) La capa final tiene K parámetros con función de activación dada por el modelo logístico[**]

$$\sigma(v) = \Big(\frac{\exp\{v_k\}}{\sum_{l=1}^K \exp\{v_l\}}\Big)_{k=1}^K.$$

(iii) Las hipótesis $h(x) \in \Delta_K$ denotan probabilidades sobre las etiquetas $y = k \in [K]$ que corresponden a clases, y la función de pérdida es el negativo de la log-verosimilitud:

$$\ell(h(x), y) = -\ln(h_y(x)).$$

(iv) Vamos a sumar un término de regularización cuadrática a la función objetivo; es decir, dado un parámetro de regularización $\lambda > 0$, buscamos minimizar

$$\mathcal{L}_\lambda = \mathbb{E}_{(X,Y)}[\ell(h(X), Y)] + \lambda\|(W, b)\|_2^2$$

donde $\|(W, b)\|_2^2$ es la suma de los cuadrados de todos los parámetros del modelo.

Notar que este modelo es una extensión del modelo de regresión logística estudiado en la tarea anterior.

Algoritmo 1 Método de Gradiente Estocástico con Minibatch

Input: Modelo inicial $w^0 \in \mathbb{R}^d$, límite de tiempo $T \in \mathbb{N}$, paso $\eta > 0$, tamaño de batch m, precisión $\epsilon > 0$
while true **do**
  Muestrear $M \sim \text{Unif}\binom{n}{m}$
  $g(w) \leftarrow \nabla f_M(w) := \frac{1}{m}\sum_{i\in M} \nabla f(w)$
  if $\|g(w)\|_2 \leq \epsilon$ o tiempo$> T$ **then return** w
  else $w \leftarrow w - \eta g(w)$
  end if
end while

[**] A diferencia del ejercicio anterior, en este caso no usamos la identidad como última función de activación, y más aún usamos una función de activación vectorial $\sigma : \mathbb{R}^{p_1} \mapsto \mathbb{R}^K$. Las elecciones hechas en el ejercicio anterior eran sólo con el propósito de simplificar el análisis.

Para ejecutar su método de gradiente estocástico, debe evaluar la pérdida sobre un conjunto de datos, así como su gradiente. Todas estas operaciones corresponden a rutinas de álgebra lineal que debe implementar en su código. Puede, por ejemplo, guiarse con el siguiente material que explica cómo implementar estas 2 operaciones: `https://ljvmiranda921.github.io/notebook/2017/02/17/artificial-neural-networks/`.

Pregunta 4. Implemente la red neuronal propuesta, y ejecútela en los conjuntos de datos MNIST (usado en la tarea anterior) y CIFAR-10 (`https://www.cs.toronto.edu/~kriz/cifar.html`. Utilice el método de gradiente estocástico con minibatch[***]

(a) Los hiperparámetros de este modelo son:

- Paso $\eta > 0$. Vamos a escoger $\eta = 10^{-i}$, con $i = 0, \ldots, 4$
- Tamaño de batch $m \in [n]$. Vamos a escoger $m = 10^k$, con $k = 1, 2, 3$.
- Ancho de la capa oculta, $p_1 \in \mathbb{N}$. Vamos a escoger 10^l, con $l = 2, 3, 4$.
- Parámetro de regularización $\lambda = 0, 10^{-2}, 10^{-1}$.

Para cada hiperparámetro determine un límite de un minuto (en conjunto con algún otro criterio de parada; por ejemplo, que la norma del gradiente en el batch sea $\leq \epsilon = 10^{-3}$) para la duración del algoritmo.

(b) Utilice una validación cruzada con el conjunto de validación para escoger un modelo. Grafique en un mapa de calor (para paso y tamaño de batch fijos a su elección) los errores de validación para las elecciones de ancho de capa y regularización.

(c) Reporte los tiempos de ejecución, y en el caso de MNIST compare sus resultados con los obtenidos en la tarea anterior.

(d) Grafique también para los hiperparámetros seleccionados las curva de riesgo de validación y testeo para los modelos obtenidos por SGD.

[***]El batch (o lote) refiere al conjunto de datos utilizados para estimar el gradiente de la prérdida empírica en cada iteración.