UNIVERSIDAD
DE GRANADA

Doble Grado en Ingeniería Informática y Matemáticas

Trabajo Fin de Grado

Fundamentos del Deep Learning y desarrollo de un modelo de Aprendizaje Federado para la diagnosis de COVID-19 a partir de radiografías de tórax

Autor
Francisco Miguel Castro Macías

Tutor
Francisco Herrera Triguero

Cotutora
Siham Tabik

Granada, Septiembre de 2020

To mum and dad...

Resumen

Your resumen text goes here.

Abstract

Abstract goes here

Yo, **Francisco Miguel Castro Macías**, alumno de la titulación Doble Grado en Ingeniería Informática y Matemáticas de la **Facultad de Ciencias** y de la **Escuela Técnica Superior De Ingenierías Informática y de Telecomunicación** de la **Universidad de Granada**, con DNI XXXXXXXXX, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca de ambos centros para que pueda ser consultada.

Granada, 3 de septiembre de 2020

F. M. Castro Macías

D. **Francisco Herrera Triguero**, Profesor del departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***Título***, ha sido realizado bajo su supervisión por **mi nombre**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expido y firmo el presente informe en Granada a 8 de julio de 2016.

profesor

Agradecimientos

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Loquesea

Theorems can easily be defined
[**latexcompanion**]

Teorema 1.1. *Let f be a function whose derivative exists in every point, then f is a continuous function.*

Teorema 1.2 (Pythagorean theorem)**.** *This is a theorema about right triangles and can be summarised in the next equation*

$$x^2 + y^2 = z^2$$

And a consequence of theorem 1.2 is the statement in the next corollary.

Corolario 1.3. *There's no right rectangle whose sides measure 3cm, 4cm, and 6cm.*

You can reference theorems such as 1.2 when a label is assigned.

Lema 1.4. *Given two line segments whose lengths are a and b respectively there is a real number r such that $b = ra$.*

Capítulo 2

Objetivos

A Stop on the Salt Route 1000 B.C. As they rounded a bend in the path that ran beside the river, Lara recognized the silhouette of a fig tree atop a nearby hill. The weather was hot and the days were long. The fig tree was in full leaf, but not yet bearing fruit. Soon Lara spotted other landmarks—an outcropping of limestone beside the path that had a silhouette like a man's face, a marshy spot beside the river where the waterfowl were easily startled, a tall tree that looked like a man with his arms upraised. They were drawing near to the place where there was an island in the river. The island was a good spot to make camp. They would sleep on the island tonight. Lara had been back and forth along the river path many times in her short life. Her people had not created the path—it had always been there, like the river—but their deerskin-shod feet and the wooden wheels of their handcarts kept the path well worn. Lara's people were salt traders, and their livelihood took them on a continual journey. At the mouth of the river, the little group of half a dozen intermingled families gathered salt from the great salt beds beside the sea. They groomed and sifted the salt and loaded it into handcarts. When the carts were full, most of the group would stay behind, taking shelter amid rocks and simple lean-tos, while a band of fifteen or so of the heartier members set out on the path that ran alongside the river. With their precious cargo of salt, the travelers crossed the coastal lowlands and traveled toward the mountains. But Lara's people never reached the mountaintops; they traveled only as far as the foothills. Many people lived in the forests and grassy meadows of the foothills, gathered in small villages. In return for salt, these people would give Lara's people dried meat, animal skins, cloth spun from wool, clay pots, needles and scraping tools carved from bone, and little toys made of wood. Their bartering done, Lara and her people would travel back down the river path to the sea. The cycle would begin again. It had always been like this. Lara knew no other life. She traveled back and forth, up and down the river path.

No single place was home. She liked the seaside, where there was always fish to eat, and the gentle lapping of the waves lulled her to sleep at night. She was less fond of the foothills, where the path grew steep, the nights could be cold, and views of great distances made her dizzy. She felt uneasy in the villages, and was often shy around strangers. The path itself was where she felt most at home. She loved the smell of the river on a hot day, and the croaking of frogs at night. Vines grew amid the lush foliage along the river, with berries that were good to eat. Even on the hottest day, sundown brought a cool breeze off the water, which sighed and sang amid the reeds and tall grasses. Of all the places along the path, the area they were approaching, with the island in the river, was Lara's favorite. The terrain along this stretch of the river was mostly flat, but in the immediate vicinity of the island, the land on the sunrise side was like a rumpled cloth, with hills and ridges and valleys. Among Lara's people, there was a wooden baby's crib, suitable for strapping to a cart, that had been passed down for generations. The island was shaped like that crib, longer than it was wide and pointed at the upriver end, where the flow had eroded both banks. The island was like a crib, and the group of hills on the sunrise side of the river were like old women mantled in heavy cloaks gathered to have a look at the baby in the crib—that was how Lara's father had once described the lay of the land. Larth spoke like that all the time, conjuring images of giants and monsters in the landscape. He could perceive the spirits, called numina, that dwelled in rocks and trees. Sometimes he could speak to them and hear what they had to say. The river was his oldest friend and told him where the fishing would be best. From whispers in the wind he could foretell the next day's weather. Because of such skills, Larth was the leader of the group. "We're close to the island, aren't we, Papa?" said Lara. "How did you know?" "The hills. First we start to see the hills, off to the right. The hills grow bigger. And just before we come to the island, we can see the silhouette of that fig tree up there, along the crest of that hill." "Good girl!" said Larth, proud of his daughter's memory and powers of observation. He was a strong, handsome man with flecks of gray in his black beard. His wife had borne several children, but all had died very young except Lara, the last, whom his wife had died bearing. Lara was very precious to him. Like her mother, she had golden hair. Now that she had reached the age of childbearing, Lara was beginning to display the fullness of a woman's hips and breasts. It was Larth's greatest wish that he might live to see his own grandchildren. Not every man lived that long, but Larth was hopeful. He had been healthy all his life, partly, he believed, because he had always been careful to show respect to the numina he encountered on his journeys. Respecting the numina was important. The numen of the river could suck a man under and drown him. The

numen of a tree could trip a man with its roots, or drop a rotten branch on his head. Rocks could give way underfoot, chuckling with amusement at their own treachery. Even the sky, with a roar of fury, sometimes sent down fingers of fire that could roast a man like a rabbit on a spit, or worse, leave him alive but robbed of his senses. Larth had heard that the earth itself could open and swallow a man; though he had never actually seen such a thing, he nevertheless performed a ritual each morning, asking the earth's permission before he went striding across it. "There's something so special about this place," said Lara, gazing at the sparkling river to her left and then at the rocky, tree-spotted hills ahead and to her right. "How was it made? Who made it?" Larth frowned. The question made no sense to him. A place was never made, it simply was. Small features might change over time. Uprooted by a storm, a tree might fall into the river. A boulder might decide to tumble down the hillside. The numina that animated all things went about reshaping the landscape from day to day, but the essential things never changed, and had always existed: the river, the hills, the sky, the sun, the sea, the salt beds at the mouth of the river. He was trying to think of some way to express these thoughts to Lara, when a deer, drinking at the river, was startled by their approach. The deer bolted up the brushy bank and onto the path. Instead of running to safety, the creature stood and stared at them. As clearly as if the animal had whispered aloud, Larth heard the words "Eat me." The deer was offering herself. Larth turned to shout an order, but the most skilled hunter of the group, a youth called Po, was already in motion. Po ran forward, raised the sharpened stick he always carried and hurled it whistling through the air between Larth and Lara. A heartbeat later, the spear struck the deer's breast with such force that the creature was knocked to the ground. Unable to rise, she thrashed her neck and flailed her long, slender legs. Po ran past Larth and Lara. When he reached the deer, he pulled the spear free and stabbed the creature again. The deer released a stifled noise, like a gasp, and stopped moving. There was a cheer from the group. Instead of yet another dinner of fish from the river, tonight there would be venison.

Capítulo 3

Aprendizaje Automático

De acuerdo con Tom Mitchell, el Aprendizaje Automático consiste en el estudio de algoritmos que mejoran automáticamente a través de la experiencia.

3.1. El problema del aprendizaje

Cuando decimos que un algoritmo aprende, ¿qué entendemos por aprender?

Definición. *Se dice que un algoritmo o programa $\mathcal{A}$ aprende de la experiencia E respecto a alguna tarea T y una medida de rendimiento P, si su rendimiento en la tarea T de acuerdo a P mejora con la experiencia E.*

Podemos formular el problema del aprendizaje en los siguientes términos. Sean $\mathcal{X}, \mathcal{Y}$ dos conjuntos que llamaremos espacio de entrada y de salida. Sea $\mathcal{D}$ el conjunto de datos o ejemplos.

Queremos aproximar una función objetivo $f : \mathcal{X} \mapsto \mathcal{Y}$ de acuerdo a una medida de error $L : \mathcal{H} \mapsto \mathbb{R}$. Nuestro algoritmo $\mathcal{A}$ usará $\mathcal{D}$ para elegir una función $g : \mathcal{X} \mapsto \mathcal{Y}$ de un conjunto de hipótesis $\mathcal{H}$ que aproxime a f.

La tarea T se corresponde con encontrar la función f, la experiencia E con el conjunto de datos $\mathcal{D}$, y la medida de rendimiento con la medida de error L.

El problema al que nos enfrentamos en este trabajo es un **problema de clasificación supervisado**. Veamos a qué nos referimos con esto.

3.1.1. Tipos de aprendizaje

En función del conjunto de datos $\mathcal{D}$ del que aprende nuestro algoritmo encontramos dos grandes clases de aprendizaje:

- **Aprendizaje supervisado:** cada ejemplo está etiquetado con el resultado que el algoritmo debería de producir. Es decir, $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$ y se busca encontrar una relación entre la etiqueta y el ejemplo.

- **Aprendizaje no supervisado:** los datos no están etiquetados. Ahora $\mathcal{D} \subset \mathcal{X}$ y pretendemos identificar propiedades de la estructura del conjunto de datos.

3.1.2. Problema de clasificación

El aprendizaje automático puede aplicarse para resolver multitud de problemas: clasificación, regresión, predicción de series temporales, detección de anomalías, generación de nuevos datos, etc.

En un problema de clasificación el espacio de salida consiste en una serie de categorías o clases y queremos que nuestro algoritmo asigne a cada ejemplo una clase de forma correcta. Es decir, $\mathcal{Y} = \{1, \ldots, C\}$ con $C \in \mathbb{N}$. Si $C = 2$, tenemos un problema de clasificación binaria.

De ahora en adelante, nos centraremos en el aprendizaje supervisado. Concretamente, hablaremos de problemas de clasificación.

3.2. Factibilidad del aprendizaje

El algoritmo de aprendizaje usa el conjunto de entrenamiento $\mathcal{D}$, que en la práctica será un conjunto finito de ejemplos, para aprender la función objetivo. ¿Por qué un conjunto limitado de datos puede revelar información suficiente para lograr nuestro objetivo? Para responder a esta pregunta tendremos que añadir algunas hipótesis adicionales a la formulación del problema. Escribamos $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ y supongamos:

H1. Tras fijar el espacio $\mathcal{H}$, los ejemplos $x_1, \ldots, x_N$ han sido extraídos de forma independiente de acuerdo a una distribución de probabilidad P sobre el espacio de entrada $\mathcal{X}$.

La función objetivo f es completamente desconocida. Lo que obtenemos como resultado del aprendizaje es una función g que aproxima a f. Para medir cómo de buena es esta aproximación, recurrimos a la función de pérdida ℓ. En la situación más general, tendremos una función $\ell : \mathcal{X} \times \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$, tal que $\ell(h, x, y)$ mide el error que cometemos al asignar a un ejemplo x la etiqueta $h(x)$ cuando su etiqueta

correcta es y. Definimos

$$L(h) = \mathbb{E}_{X\sim P}[\ell(h, X, f(X))] \quad \forall h \in \mathcal{H}$$

y queremos resolver:

$$\text{Encontrar } g \in \mathcal{H} \text{ tal que } L(g) = \min_{h\in\mathcal{H}} L(h)$$

No tenemos acceso a la función f ni tampoco a la distribución P. Por tanto, resolver el anterior problema será inviable en la mayor parte de las situaciones. Una forma natural de proceder es definir

$$L_{in}(h, \mathcal{D}) = \frac{1}{N} \sum_{(x_n,y_n)\in\mathcal{D}} \ell(h, x_n, y_n) \quad \forall h \in \mathcal{H}$$

y, una vez fijado $\mathcal{D}$, minimizar L_{in} en lugar de L:

$$\text{Encontrar } g \in \mathcal{H} \text{ tal que } L_{in}(g) = \min_{h\in\mathcal{H}} L_{in}(h)$$

Lo que vamos a probar es que reducir L_{in} conlleva reducir L, luego podemos aspirar a soluciones subóptimas.

Para un problema de clasificación binaria, definimos $\ell(u, v) = 1$ si $u = v$ y 0 si $u \neq v$. Vamos a omitir la dependencia de $\mathcal{D}$ al escribir L_{in} y vamos a llamar $L_{out} = L(h) = P[h(x) \neq f(x)] \quad \forall h \in \mathcal{H}$. Vamos a usar la desigualdad de Hoeffding:

Lema 3.1 (Desigualdad de Hoeffding)**.** *Sea $Z_1, \ldots, Z_N$ una muestra aleatoria simple de una variable aleatoria Z que sigue una distribución de Bernouilli con media $E[Z]$. Sea $\bar{Z}$ la media muestral. Sea ϵ un número real estrictamente positivo. Entonces se cumple que:*

$$P[|E[Z] - \bar{Z}| > \epsilon] \leq 2\epsilon^{-2\epsilon^2 N} \tag{3.1}$$

Supongamos primero que $\mathcal{H} = \{h\}$. Definamos $I : \mathcal{X} \mapsto \{0, 1\}$ como $I(x) = 1$ si $h(x) = f(x)$ y $I(x) = 0$ si $h(x) \neq f(x)$. Tenemos entonces que I es una variable aleatoria que sigue una distribución de Bernouilli, cuya esperanza se corresponde con L_{out}. Si escribimos $I_n = \ell(x_n, y_n)$, tenemos que $I_1, \ldots, I_n$ es una muestra aleatoria simple de I, y podemos aplicar (3.1) para un $\epsilon > 0$ fijo obteniendo que:

$$P[|L_{out}(h) - L_{in}(h)| > \epsilon] \leq 2\epsilon^{-2\epsilon^2 N} \tag{3.2}$$

Esta desigualdad nos dice que, fijado un margen de error ϵ, la diferencia entre

L_{in} y L_{out} queda controlada por el tamaño de la muestra. Aumentando el tamaño de N, la probabilidad de exceder el margen de error tiende a cero.

Ahora supongamos que $\mathcal{H} = \{h_1, \ldots, h_M\}$. Denotemos por g la solución aproximada que obtiene el algoritmo $\mathcal{A}$. Denotemos por $\mathbb{B}$ el evento $|L_{out}(g) - L_{in}(g)| > \epsilon$ y para cada h_m, por $\mathbb{B}_m$ el evento $|L_{out}(h_m) - L_{in}(h_m)| > \epsilon$. Se cumple que $\mathbb{B} \subset \bigcup_{m=1}^{M} \mathbb{B}_m$. Entonces:

$$P[\mathbb{B}] \leq P[\bigcup_{m=1}^{M} \mathbb{B}_m] \leq \sum_{m=1}^{M} P[\mathbb{B}_m]$$

Podemos aplicar la desigualdad (3.2) a cada término para obtener:

$$P[|L_{out}(g) - L_{in}(g)| > \epsilon] \leq 2M\epsilon^{-2\epsilon^2 N}$$

Hemos obtenido una generalización de la desigualdad (3.2) para el caso en que $\mathcal{H}$ tiene más de un elemento, y a la que se le puede aplicar un razonamiento análogo al que hacíamos antes.

Vamos a escribir la desigualdad anterior de otra forma. Llamemos $\delta = 2M\epsilon^{-2\epsilon^2 N}$, luego $\epsilon = \sqrt{\frac{1}{2N} log\left(\frac{2M}{\delta}\right)}$. Tenemos que, con probabilidad al menos $1 - \delta$, se cumple:

$$L_{out}(g) - L_{in}(g) \leq |L_{out}(g) - L_{in}(g)| \leq \epsilon$$

de donde, nuevamente con probabilidad al menos $1 - \delta$,

$$L_{out}(g) \leq L_{in}(g) + \sqrt{\frac{1}{2N} log\left(\frac{2M}{\delta}\right)} \tag{3.3}$$

Como vemos, fijada una tolerancia δ el error L_{out} queda dominado por el error dentro de la muestra, L_{in}, más un término positivo. Este término positivo converge a cero cuando el tamaño de la muestra aumenta.

3.2.1. Dimensión de Vapnik-Chervonenkis

En la práctica, la clase de funciones entre la que buscamos la solución será infinita. Para dar una respuesta en este caso, debemos recurrir a un razonamiento más técnico. Exponemos aquí brevemente las ideas básicas de la teoría de la dimensión de Vapnik-Chervonenkis.

Si $x_1, \ldots, x_N$ son N puntos de $\mathcal{X}$, cada $h \in \mathcal{H}$ genera una dicotomía que podemos

denotar por $(h(x_1), \ldots, h(x_N))$.

Definición 3.2 (Dicotomías generadas por $\mathcal{H}$)**.** *Sean $x_1, \ldots, x_N \in \mathcal{X}$, las dicotomías generadas por $\mathcal{H}$ en $x_1, \ldots, x_N$ se denotan por $\mathcal{H}(x_1, \ldots, x_N) = \{(h(x_1), \ldots, h(x_N)) \colon h \in \mathcal{H}\}$*

Definición 3.3 (Función de crecimiento)**.** *Sea $\mathcal{H}$ una clase de hipótesis. Definimos la función de crecimiento de $\mathcal{H}$ como*

$$\begin{aligned} m_{\mathcal{H}} \colon \mathbb{N} &\to \mathbb{R} \\ N &\mapsto \max_{x_1, \ldots, x_N \in \mathcal{X}} \mathcal{H}(x_1, \ldots, x_N) \end{aligned}$$

donde $|A|$ denota el cardinal de A.

Como $\mathcal{H}(x_1, \ldots, x_N) \subset \{0,1\}^N$, entonces $m_{\mathcal{H}}(N) \leq 2^N$ para cada $N \in \mathbb{N}$. Si para algún natural k ocurre que $m_{\mathcal{H}}(k) \leq 2^k$, podemos acotar la función de crecimiento por un polinomio de grado $k-1$:

Proposición 3.4. *Si $m_{\mathcal{H}}(k) < 2^k$ para algún número natural k, entonces*

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{k-1} \binom{N}{i} \quad \forall N \in \mathbb{N}$$

La anterior propiedad motiva el concepto clave de esta teoría. La dimensión de Vapnik-Chervonenkis nos permite medir cómo de grande es el conjunto de hipótesis.

Definición 3.5 (Dimensión de Vapnik-Chervonenkis)**.** *Sea $\mathcal{H}$ una clase de hipótesis. La dimensión de Vapnik-Chervonenkis o dimensión VC de $\mathcal{H}$, denotada por $d_{VC}(\mathcal{H})$, es:*

$$d_{VC}(\mathcal{H}) = \max\{N \in \mathbb{N} \colon m_{\mathcal{H}}(N) = 2^N\}$$

entendiendo que si $m_{\mathcal{H}}(N) = 2^N$ para cada $N \in \mathbb{N}$, entonces $d_{VC}(\mathcal{H}) = \infty$.

El resultado central de esta teoría, y el cual nos permitirá dar una respuesta a la pregunta que nos planteamos en esta sección, se enuncia como sigue:

Teorema 3.6 (Cota VC)**.** *Sea $\delta > 0$. Entonces:*

$$L_{out}(g) \leq L_{in}(g) + \sqrt{\frac{8}{N} log\left(\frac{4m_{\mathcal{H}}(2N)}{\delta}\right)}$$

con probabilidad al menos $1 - \delta$.

Si $d_V C(\mathcal{H})$ es finita, entonces podemos aplicar la proposición 3.4. Al tomar límite cuando N tiende a infinito, vemos que la diferencia entre L_{out} y L_{in} tiende a cero.

Como vemos, esta desigualdad guarda cierto parecido a la que obteníamos cuando $\mathcal{H}$ es finita. Al igual que en aquella, si podemos asegurar que el tamaño de la clase de hipótesis es finito (en este caso a través de la dimensión VC), podemos asegurar que la diferencia entre el error fuera y dentro de la muestra tiende a cero cuando aumenta el tamaño de esta.

3.3. Optimización: gradiente descendente

En la anterior sección hemos argumentado por qué para obtener un modelo de aprendizaje automático el problema que se resuelve es:

$$\text{Encontrar } g \in \mathcal{H} \text{ tal que } L_{in}(g) = \min_{h \in \mathcal{H}} L_{in}(h)$$

Por tanto, todos los esfuerzos se centran en minimizar una función L_{in}. En la práctica, cada hipótesis h queda completamente determinada por unos parámetros $w \in \mathbb{R}^d$. Entonces, podemos reformular el problema de la siguiente forma:

$$\text{Encontrar } w^* \in \mathbb{R}^d \text{ tal que } L_{in}(w^*) = \min_{w \in \mathbb{R}^d} L_{in}(w)$$

Nos encontramos ante un problema de optimización y nos interesa conocer métodos para resolverlo.

Vamos a estudiar uno de los métodos más efectivos. En la práctica, los algoritmos basados en el gradiente descendente obtienen muy buenos resultados. Suponiendo que la función que queremos optimizar es diferenciable, la idea del gradiente descendente consiste en partir de un punto aleatorio, y, fijándonos en el gradiente, movernos en la dirección en la que decrece la función.

Algoritmo 1: Gradiente descendente

Entrada: Función F, tasa de aprendizaje η
Inicializar w(0)
para *t = 0, 1, 2, ...* **hacer**
  $w(t+1) = w(t) - \eta \nabla F(w(t))$
fin

3.3.1. Convergencia del método

Podemos probar que, bajo ciertas hipótesis no muy restrictivas, el algoritmo llega a obtener un mínimo global. Para ello, vamos a demostrar algunos resultados previos. Fijemos una función $F\colon \mathbb{R}^d \mapsto \mathbb{R}$ tal que $F \in \mathcal{C}^1$. Denotemos por $\|\cdot\|$ a la norma euclídea.

Definición 3.7 (Función convexa)**.** *Decimos que F es convexa si*

$$F(tx + (1-t)y) \leq tF(x) + (1-t)F(y) \quad \forall x, y \in \mathbb{R}^d, t \in [0,1]$$

Definición 3.8 (Gradiente Lipschitz)**.** *Decimos que el gradiente de F es lipschitziano con constante $L > 0$ si*

$$\|\nabla F(x) - \nabla F(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^d$$

Lema 3.9. *Sean $x, y \in \mathbb{R}^d$.*

1. *Si F es convexa, cualquier mínimo local es un mínimo global.*

2. *Si F es convexa, entonces*

$$\nabla F(y)^T(x - y) \leq F(x) - F(y) \tag{3.4}$$

3. *Si el gradiente de F es lipschitziano con constante $L > 0$, entonces*

$$F(y) \leq F(x) + \nabla F(x)^T(y - x) + \frac{L}{2}\|y - x\|^2 \tag{3.5}$$

Demostración.

1. Supongamos que x^* es un mínimo local. Entonces existe $r > 0$ tal que $F(x^*) < F(x)$ para cada $x \in B(x^*, r)$. Supongamos para llegar a contradicción que existe z tal que $F(z) < F(x^*)$. Entonces, para $t \in [0,1]$:

$$F(tx^* + (1-t)z) \leq tF(x^*) + (1-t)F(z) < F(x^*)$$

   Evaluando en $t = 1$ se obtiene que $F(x^*) < F(x^*)$, lo cual no es posible.

2. Fijemos $x, y \in \mathbb{R}^d$. De la definición de convexidad, se cumple que:

$$F(y + t(x - y)) - F(y) \leq t(F(x) - F(y))$$

Diviendo por $t > 0$ ambos términos, podemos tomar límite cuando t tiende a cero y aplicar la definición de diferencial obteniendo la desigualdad buscada.

3. Definimos la función $g(t) = f(x+t(y-x))$ para cada $t \in [0,1]$. Se cumple que g es derivable con $g'(t) = \nabla F(x+t(y-x))^T(y-x)$ para cada $t \in [0,1]$. Aplicando la desigualdad de Cauchy-Schwartz y la definición de gradiente lipschiztiano:

$$\begin{aligned} g'(t) - g'(0) &= \left[\nabla F(x+t(y-x)) - \nabla F(x)\right]^T (y-x) \leq \\ &\leq \|\nabla F(x+t(y-x)) - \nabla F(x)\| \|y-x\| \leq Lt\|y-x\|^2 \end{aligned}$$

Por el teorema fundamental del cálculo:

$$\begin{aligned} F(y) = g(1) = g(0) + \int_0^1 g'(t)dt \leq g(0) + g'(0) + L\|y-x\|^2 \int_0^1 tdt = \\ = F(x) + \nabla F(x)^T(y-x) + \frac{L}{2}\|y-x\|^2 \end{aligned}$$

□

Podemos probar ya la convergencia del método del gradiente descendente. Supondremos la existencia de un punto donde se alcanza el mínimo de la función F.

Proposición 3.10. *Sea $F\colon \mathbb{R}^d \mapsto \mathbb{R}$ tal que $F \in \mathcal{C}^1$, es convexa y su gradiente es lipschitziano con constante $L > 0$. Supongamos que existe x^* tal que $F(x^*) \leq F(x)$ para todo $x \in \mathbb{R}^d$. Sea $\eta \leq \frac{1}{L}$. Sea $x_0 \in \mathbb{R}^d$ y definamos $x_k = x_{k-1} - \eta\nabla F(x_{k-1})$ para cada $k \in \mathbb{N}$. Entonces:*

$$F(x_k) - F(x^*) \leq \frac{\|x_0 - x^*\|^2}{2\eta k}$$

Demostración. Vamos a fijar $x \in \mathbb{R}^d$ y llamamos $y = x - \eta\nabla F(x)$. De la desigualdad (3.5) tenemos que:

$$F(y) \leq F(x) - \eta\|\nabla F(x)\|^2 + \frac{1}{2}L\eta^2\|\nabla F(x)\|^2$$

Usando que $\eta \leq \frac{1}{L}$, llegamos a:

$$F(y) \leq F(x) - \frac{1}{2}\eta\|\nabla F(x)\|^2$$

En particular, $F(y) \leq F(x)$, y por tanto $F(x_k) \leq F(x_i)$ para cada $i \leq k$, lo cual prueba que la actualización del gradiente descendente conlleva el decrecimiento de

la función objetivo en cada iteración. Ahora usando la desigualdad (3.4), obtenemos

$$F(x) \leq F(x^*) + \nabla F(x)^T (x - x^*)$$

Combinando las dos últimas desigualdades:

$$F(y) \leq F(x^*) + \nabla F(x)^T (x - x^*) - \frac{1}{2}\eta \|\nabla F(x)\|^2$$
$$F(y) - F(x^*) \leq \frac{1}{2\eta} \left(2\eta \nabla F(x)^T (x - x^*) - \eta^2 \|\nabla F(x)\|^2\right)$$
$$F(y) - F(x^*) \leq \frac{1}{2\eta} \left(2\eta \nabla F(x)^T (x - x^*) - \eta^2 \|\nabla F(x)\|^2 + \|x - x^*\| - \|x - x^*\|\right)$$

Observando que $\|y - x^*\|^2 = \|x - \eta \nabla F(x) - x^*\|^2 = \|x - x^*\|^2 + \eta^2 \|\nabla F(x)\|^2 - 2\eta \nabla F(x)^T (x - x^*)$, se llega a:

$$F(y) - F(x^*) \leq \frac{1}{2\eta} \left(\|x - x^*\|^2 - \|y - x^*\|^2\right)$$

Basta aplicar la desigualdad obtenida eligiendo $x = x_{i-1}$, $y = x_{i-1} - \eta \nabla F(x_{i-1})$ de la siguiente forma:

$$\begin{aligned} k\left(F(x_k) - F(x^*)\right) &\leq \sum_{i=1}^{k} \left(F(x_i) - F(x^*)\right) \leq \sum_{i=1}^{k} \frac{1}{2\eta} \left(\|x_{i-1} - x^*\|^2 - \|x_i - x^*\|^2\right) = \\ &= \frac{1}{2\eta} \left(\|x_0 - x^*\|^2 - \|x_k - x^*\|^2\right) \leq \\ &\leq \frac{1}{2\eta} \left(\|x_0 - x^*\|^2\right) \end{aligned}$$

□

El resultado que acabamos de probar nos garantiza que, si existe un punto en el que la función objetivo alcanza su valor mínimo, el algoritmo del gradiente descendente converge.

3.3.2. Aplicación en el aprendizaje

Recordemos que las funciones de pérdida se definen como:

$$L_{in}(w) = \frac{1}{N} \sum_{n=1}^{N} \ell(w, x_n, y_n)$$

Asumiendo que ℓ es diferenciable, su gradiente se expresará como:

$$\nabla L_{in}(w) = \frac{1}{N}\sum_{n=1}^{N} \nabla_w \ell(w, x_n, y_n)$$

Si aplicamos el algoritmo de gradiente descendente, en cada iteración tenemos que calcular $\nabla_w \ell$ sobre cada ejemplo de entrenamiento. Lo que ocurre en la práctica, y más en el ámbito del Deep Learning es que el cálculo de los gradientes conlleva también un gran número de operaciones. Esto, sumado a que los conjuntos de datos son muy grandes, hace que la eficiencia sea un aspecto a tener en cuenta.

Por este motivo se considera una variante llamada **gradiente descendente estocástico (SGD)** que en la práctica consigue muy buenos resultados. En esta aproximación, en lugar de usar en cada iteración todo el conjunto de datos, se elige un conjunto reducido de muestras (minibatch) de forma aleatoria y se hacen las actualizaciones de esta forma.

Algoritmo 2: Gradiente descendente estocástico

Entrada: Función ℓ, tasa de aprendizaje η, conjunto $\mathcal{D}$
Inicializar w(0)
para *t = 0, 1, 2, ...* **hacer**
  Escoger un minibatch $\{x_1, \ldots, x_M\}$
  Calcular $g(t) = \sum_{m=1}^{M} \nabla_w \ell(w(t), x_m, y_m)$
  $w(t+1) = w(t) - \eta g(t)$
fin

Por otra parte, debemos realizar una puntualización sobre las hipótesis bajo las cuales el gradiente descendente tiene garantizada su convergencia. En la práctica, las funciones que intentamos minimizar en aprendizaje automático y en Deep Learning no son globalmente convexas y pueden tener muchos mínimos locales que no son globales, así como puntos donde el gradiente se anula (puntos de silla) rodeados de regiones planas.

Al aplicar el gradiente descendente a estas funciones, obtenemos soluciones en los que el valor de la función de pérdida es muy bajo, pero no necesariamente óptimo.

3.3.3. Variantes

Para usar el gradiente descendente hemos de elegir el criterio de parada y la tasa de aprendizaje. La proposición 3.10 pone de manifiesto la importancia de elegir un valor correcto para η. Por otra parte, elegir un criterio de parada erróneo puede hacer que el algoritmo oscile alrededor de un mínimo, obteniendo soluciones cada

vez peores.

Existen algunas variantes de SGD que adaptan estos parámetros conforme avanzan las iteraciones. Se ha comprobado empíricamente que en muchos casos su aplicación conduce a mejores resultados. A continuación, describimos brevemente la idea en que se basan algunos de ellos.

SGD con momento

Es una técnica para acelerar la convergencia del gradiente descendente. Se basa en acumular en cada iteración t un vector de velocidades $v(t)$ en las direcciones en las que más está descendiendo el valor de la función objetivo:

$$v(t+1) = \mu v(t) - \eta \nabla F(w(t))$$
$$w(t+1) = w(t) + v(t+1)$$

siendo $\eta > 0$ la tasa de aprendizaje y $\mu \in [0, 1]$ el coeficiente de momento.

Adagrad y RMSprop

Adagrad adapta la tasa de aprendizaje a los parámetros w. En lugar de mantener la misma para todos, a aquellos que tienen mayores derivadas parciales les corresponde una tasa más baja. Si $w \in \mathbb{R}^d$ definimos:

$$G(0)_i = [\nabla F(w(0))_i]^2$$
$$G(t+1)_i = G(t)_i + [\nabla F(w(t+1))_i]^2$$

La regla de actualización queda:

$$w(t+1)_i = w(t)_i - \frac{\eta}{\sqrt{G(t)_i + \epsilon}} \nabla F(w(t))_i$$

siendo $\eta > 0$ la tasa de aprendizaje y $\epsilon > 0$ un término pequeño para evitar la división por cero.

El mayor inconveniente de Adagrad es que las componentes de G en cada iteración pueden crecer indefinidamente, haciendo que las actualizaciones de los parámetros no tengan efecto. RMSProp intenta solucionar esto definiendo:

$$G(t+1)_i = \alpha G(t)_i + (1 - \alpha)[\nabla F(w(t+1))_i]^2$$

para $\alpha \in [0, 1]$

Adam

Capítulo 4

Deep Learning

4.1. Introduction

Capítulo 5

Aprendizaje Federado

Apéndice A

Appendix Title