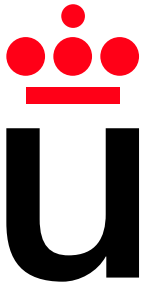

Universidad
Rey Juan Carlos

Escuela Técnica Superior
de Ingeniería Informática

Grado en Matemáticas

Curso 2023-2024

Trabajo de Fin de Grado

SOBRE ECUACIONES DIFERENCIALES ORDINARIAS EN REDES NEURONALES

Autora: Laura Medina Henche

Tutor: Emanuele Schiavi

Agradecimientos

A mi hermana y mis padres, que siempre han estado ahí de manera incondicional, contra viento y marea. Gracias por aguantar mis explicaciones interminables sobre materias que no llegabais a entender y por apoyarme allá donde voy.

A mis abuelos, que me han acogido como su hija durante todos mis años universitarios, aguantando jornadas de estudio de sol a sol y cuidándome sin descanso. Gracias por ser hogar lejos de casa.

Al resto de familiares, amigos y compañeros que me han apoyado a lo largo de este arduo recorrido. Gracias por ayudarme a desconectar cuando más lo he necesitado y por quererme tal y como soy. En especial a Paula, Irene, Mónica y Carlos, los mejores amigos que podría tener.

También a todos mis profesores, por enseñarme y compartir conmigo sus conocimientos. Gracias en especial a Emanuele, por guiarme en el desarrollo de este trabajo de fin de grado. La persona que hoy escribe estas líneas es un reflejo de vuestro esfuerzo, consideración y amor por el trabajo. Ha sido un honor aprender de vosotros, gracias.

Y a Isabel y Montse, que me supieron transmitir el amor por las matemáticas hasta poder hacer de su pasión la mía. Gracias por abrirme los ojos a una materia maravillosa que a partir de hoy puedo llamar, con orgullo, mía.

Resumen

El éxito de la inteligencia artificial y su rápido desarrollo, a través del diseño de nuevas arquitecturas y algoritmos para redes neuronales, está promoviendo un creciente interés por la comprensión y mejora de los procesos y algoritmos utilizados.

En este trabajo estudiaremos los fundamentos matemáticos en los que se basa el diseño y desarrollo de las recientemente propuestas redes neuronales definidas por ecuaciones diferenciales ordinarias, de acrónimo ODENets[1] [1].

Se trata de redes neuronales *continuas* cuya discretización, a través de un método de Euler explícito recupera la formulación de unas redes neuronales *discretas* conocidas como redes residuales o ResNets[2].

Dado que los parámetros óptimos de la red minimizan una función de pérdida, podemos ver la red como un problema de minimización con restricciones de un funcional de energía en el marco de la teoría del control óptimo de los sistemas dinámicos.

La implicación fundamental de este marco continuo es que la optimización se realiza ahora en espacios infinitos dimensionales por lo cual se aplica el cálculo variacional para el planteamiento y resolución del problema de optimización.

Mediante el cálculo del hamiltoniano del sistema se escriben las ecuaciones de Euler-Lagrange del problema de optimización como un sistema de EDO de primer orden. El método del adjunto y el principio del máximo de Pontryagin permitirán el tratamiento matemático y la resolución del problema.

Palabras clave:

- Optimización con restricciones
- Multiplicadores de Lagrange
- Método del adjunto
- Principio del máximo de Pontryagin
- ResNet
- ODENet

[1] Ordinary Differential Equations Networks
[2] Residual Networks

Objetivos

El objetivo general de este trabajo es comprender las matemáticas en las que se fundamenta la implementación de una red neuronal creada mediante la resolución de ecuaciones diferenciales ordinarias, como se explica en *Neural Ordinary Differential Equations*, que obtuvo uno de los premios al mejor artículo en NIPS 2018 [1].

Como objetivos específicos, podemos exponer los siguientes:

- Desarrollar los conocimientos adquiridos a lo largo del plan de estudios en asignaturas como cálculo, estadística, EDO y EDP.
- Aplicar los conceptos y teoría de la optimización con restricciones, multiplicadores de Lagrange, método del adjunto y principio del máximo de Pontryagin a la inteligencia artificial y el diseño de redes.
- Estudiar y aplicar técnicas de optimización con restricciones para la resolución de problemas de optimización.
- Conectar teoría, cálculo simbólico, cálculo numérico, resolución, aplicación y representación de problemas de optimización.
- Utilizar programas de cálculo simbólico como Matlab y Python para la resolución de algoritmos y la representación gráfica de los resultados.
- Aplicar métodos numéricos para resolver las EDOs que aparecen en los distintos problemas.
- Definir, estudiar, resolver e interpretar un modelo para la resolución del problema de aterrizaje lunar.
- Realizar los pasos conceptuales para la comprensión de los algoritmos de aprendizaje y redes neuronales expuestos en *Neural Ordinary Differential Equations* [1].

Índice de contenidos

Índice de figuras

Índice de códigos

Introducción

La inteligencia artificial es una rama de la ciencia que se ocupa de crear máquinas inteligentes a través del diseño de algoritmos capaces de detectar patrones y dinámicas en los datos de un problema.

Tras el entrenamiento (*training*) con un conjunto de datos etiquetados, pueden hacer inferencias y predecir el comportamiento del sistema que modela el problema frente a la entrada de un nuevo *input* o conjunto de datos nunca visto por la red.

El aprendizaje automático (*machine learning*, ML) es una rama de la inteligencia artificial que incluye métodos y algoritmos para crear modelos de datos automáticamente. Los métodos de ML pueden ser supervisados, semisupervisados y no supervisados, dependiendo de la tarea final (*downstream task*) y de la cantidad y calidad de los datos disponibles. Si hay datos etiquetados (por expertos), se utilizan métodos supervisados que aprenden las relaciones (características o *features*) que permiten clasificar un nuevo dato. Si los datos son en crudo (*raw data*), se utilizan métodos no supervisados, que permiten entender patrones y tendencias del conjunto de datos (*dataset*).

El ML es por tanto una tecnología para el análisis de datos y la toma de decisiones con múltiples aplicaciones en el ámbito social e industrial. Se ocupa de los patrones de aprendizaje y las relaciones entre los datos de entrenamiento. Al contrario de los sistemas que ejecutan tareas siguiendo reglas explícitas, un sistema de ML aprende de la experiencia.

El aprendizaje profundo (*deep learning*) es una técnica de ML que construye redes neuronales artificiales para simular la estructura y funcionamiento del cerebro humano. Las redes neuronales se basan en la conexión de nodos llamados neuronas artificiales que reciben datos, los procesan y producen una salida, llamada activación (neuronal).

Los nodos se organizan en capas (*layers*), por lo que los datos pasan a la primera capa, que realimenta a la segunda, y así sucesivamente hasta llegar a la capa de salida, que nos proporciona una solución a nuestro problema, típicamente a través de una clasificación o una regresión. La salida de cada capa se obtiene a través de la aplicación de unas funciones no lineales que "activan", o no, una neurona.

Dependiendo del número de neuronas, nuestra red puede ser más o menos profunda (es decir, con más o menos capas). Se considera un número mínimo de tres capas para que una red se pueda considerar profunda. La estructura mínima de una red profunda se compone al menos de una capa de entrada (*input*), una capa interior o capa oculta (*hidden layer*) y una capa de salida (*output*). Al añadir más capas ocultas se añade profundidad a la red.

A cada neurona se le aplica un sesgo o *bias*, que se suma al producto de los datos de entrada por una matriz de pesos; y a la salida de cada neurona puede haber una función no lineal denominada de activación que puede modificar el valor de salida de la misma, bien normalizándolo, aumentándolo o inhibiéndolo, entre otros. En la figura 1 podemos ver un ejemplo de una red neuronal profunda.

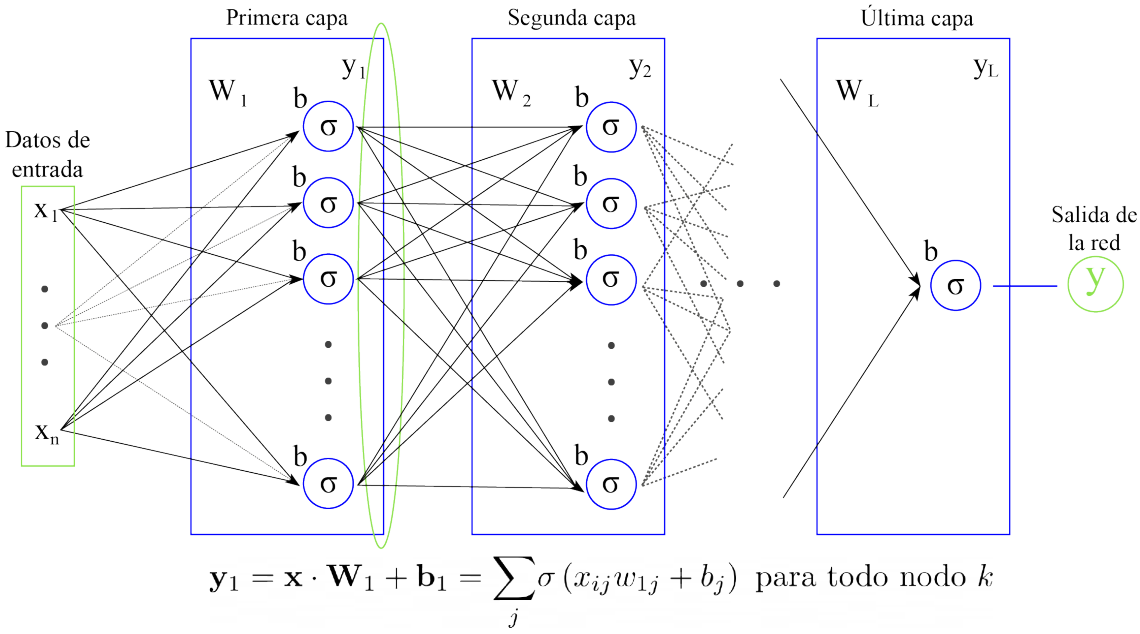

Figura 1: Ejemplo de red neuronal profunda.

Podemos formalizar las salidas (activaciones) de cada capa de la red en la forma:

$$\mathbf{y}_i = \sigma(W_i \mathbf{X} + \mathbf{b}_i), \qquad i = 1...L$$

En la figura 1 podemos ver que los datos de entrada de la red se representan mediante un vector o matriz $\mathbf{X}$, y la salida mediante y_L, que puede ser tanto un vector como un escalar; los pesos se corresponden con los vectores $\mathbf{W}_i$, $i \in [1, L]$; los sesgos se corresponden con los vectores $\mathbf{b}_i$, $i \in [1, L]$, y σ representa la función de activación. En la figura 1, la red neuronal tiene L capas y es importante notar que los vectores $\mathbf{W}_i$ y $\mathbf{b}_i$ tienen tantas componentes como neuronas hay en la capa i. La salida de cada capa i se representa como $\mathbf{y}_i$. En la fórmula, w_{1j} representa el peso del vector $\mathbf{W}_1$ que multiplica la entrada j y la propaga al siguiente nodo.

Lo que se busca en una red neuronal es encontrar la salida óptima, es decir, los parámetros óptimos, mediante la actualización de los pesos. Esta búsqueda se denomina *aprendizaje*, ya que la red parte de unos pesos y va modificándolos hasta obtener una respuesta lo suficientemente precisa. Para que la red pueda entrenar y aprender es necesario que todas las operaciones realizadas por el algoritmo sean diferenciables, asegurando la existencia de las derivadas parciales de la función de pérdida en función de los parámetros de la red. El cálculo de las derivadas se realiza aplicando la regla de la cadena mediante un proceso de derivación definido de propagación retrógrada (*backpropagation*).

ResNets

Una red neuronal residual es una red que, además de las típicas conexiones entre capas consecutivas, incluye conexiones entre capas no consecutivas, normalmente saltándose dos o tres capas. Estos "atajos" sirven para evitar la anulación del gradiente (*vanishing gradient*), que se produce cuando el gradiente se vuelve muy pequeño y los pesos no llegan a cambiar su valor con la actualización de los mismos (y, por tanto, se estanca el aprendizaje).

Las ResNets surgieron para intentar mejorar el entrenamiento de las redes neuronales profundas, ya que, como He et al. comprobaron en *Deep residual learning for image recognition* [2], al añadir más capas se incrementaba el error tanto de entrenamiento como de test, lo cual atribuyeron al *overfitting*[3].

Estas redes construyen transformaciones complejas mediante la composición de una secuencia de transformaciones a un estado oculto h_t que evoluciona según el esquema numérico:

$$h_{t+1} = h_t + f(h_t, \theta_t) \tag{1}$$

con $t \in \{0...T\}$ y $h_t \in \mathbb{R}^D$. La variable θ_t denota al valor de los parámetros de la red en un instante (o capa) y hace que el esquema numérico descrito en (1) sea paramétrico. Nótese que la notación típica en redes es la de h_l siendo l el *layer* o capa de la red neuronal.

Podemos observar que estas actualizaciones realizadas de manera iterativa pueden verse como una discretización de Euler de una transformación continua $y' = f(t, y)$. De hecho, aplicando el método de Euler explícito para la discterización de y' con paso h se obtiene:

$$y_{n+1} = y_n + hf(t_n, y_n) \tag{2}$$

donde y_n define el estado del sistema y f es la función que define el problema.

[3]Sobreajuste del modelo a los datos de entrenamiento. Ocurre durante la fase de entrenamiento, cuando la red "aprende de memoria"sobre los datos y deja de poder aproximar correctamente los nuevos datos de entrada.

Ecuaciones diferenciales ordinarias neuronales

Las ecuaciones diferenciales ordinarias neuronales nacen al modelar una red continua (ODENet) cuya discretización mediante el método de Euler explícito recupera la formulación de una ResNet.

En una ODENet se parametriza la derivada de la capa oculta utilizando una red neuronal. La salida se computa con un método de resolución de ecuaciones, por ejemplo, el método de Euler o el de Runge Kutta. Otros métodos, adaptativos, también han sido propuestos.

Si en una red que utiliza las transformaciones mencionadas anteriormente añadimos muchas más capas y vamos haciendo los pasos cada vez más pequeños, en el límite podemos parametrizar la dinámica continua de las capas ocultas utilizando una EDO especificada por una red neuronal:

$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta) \tag{3}$$

Es importante observar que no tenemos que resolver una EDO dada fija, sino que la red neuronal tiene que *aprender* los parámetros óptimos de una función paramétrica f a partir de los datos. En esto consistirá el entrenamiento de la red en un conjunto de datos etiquetados. La red entrenada, es decir, la EDO (3) con parámetros óptimos, será luego utilizada para *inferir* nuevos comportamientos a partir de nuevos datos.

El aspecto técnico más complejo en este tipo de redes es llevar a cabo la *backpropagation*[4] en el integrador numérico (ODE *solver*). Para esto, los autores del artículo *Neural ordinary differential equations* [1] decidieron tratar el ODE *solver* como una caja negra y calcular los gradientes utilizando el método del adjunto.

Con esto seguimos yendo hacia atrás, pero nos ahorramos el coste de memoria que supone la *backpropagation*. En efecto, durante el proceso de propagación retrógrada es necesario almacenar los valores de las activaciones calculados durante el paso de una capa a la otra durante la propagración hacia delante (*feed forward*) de la red, lo que es mucho más costoso, en términos de memoria, que el almacenamiento de los parámetros de la red. Este problema se conoce con el nombre de *memory bottleneck in ResNets*, es decir, la memoria es el cuello de botella en las redes neuronales residuales. Estas consideraciones se pueden encontrar en *Do residual neural networks discretize neural ordinary differential equations?* de Sanders et al. [3].

La idea es aproximar las activaciones utilizando un esquema numérico explícito de Euler (2) invirtiendo el tiempo para la propagación retrógrada. Podemos simular dinámicas continuas y cambiar de algoritmo (adaptativo) para acceder a los estados latentes del sistema (*output* de cada capa).

[4]Método de cálculo del gradiente para calcular los pesos en una red neuronal, propagando las salidas de error hacia atrás, de forma retrógrada, para el cálculo de las derivadas del funcional de pérdida con respecto a los parámetros de la red.

El *output* de la red siempre verifica un principio de optimización: sus parámetros son óptimos, ya que minimizan una función de pérdida. Es decir, podemos ver nuestra red como un problema de **minimización con restricciones**. Se trata, por tanto, de minimizar una función de pérdida, siendo los estados latentes de la red soluciones de una EDO vectorial paramétrica.

Fundamentos matemáticos previos

Para poder abordar las matemáticas avanzadas utilizadas para el diseño e implementación eficiente de las ODENets, es necesaria una base matemática sobre cálculo multivariable, ecuaciones diferenciales ordinarias, métodos numéricos y teoría de la optimización. Estos conocimientos facilitan la comprensión de conceptos más complejos, como los sistemas hamiltonianos y el principio del máximo de Pontryagin, que serán abordados y ejemplificados en la parte final de la memoria.

La optimización sin restricciones se estudia durante el grado en las asignaturas de estadística y cálculo, en las cuales se definen los operadores diferenciales multivariables de derivación parcial, gradiente, matriz jacobiana y hessiana. Por su parte, las ecuaciones diferenciales ordinarias, las ecuaciones en derivadas parciales y los métodos numéricos se consideran en las respectivas asignaturas.

Junto con la capacidad de análisis adquirida a lo largo del grado, estos conceptos permiten la comprensión de la materia abordada en esta memoria.

Como complemento a la teoría de optimización con restricciones se ha incluido un anexo sobre optimización sin restricciones (anexo A) para repasar los conocimientos adquiridos a lo largo de la carrera en algunas asignaturas, como estadística. Sirven de base para el primer capítulo del trabajo, que excede lo estudiado durante el grado.

Para la primera parte del trabajo se han utilizado como base el libro *Numerical Optimization*, de J. Nocedal y S. J. Wright [4], los apuntes del máster de visión artificial de E. Schiavi e I. Ramírez [5] y el libro *Convex Optimization*, de S. Boyd y L. Vandenberghe [6].

Sobre la notación: En esta memoria se ha considerado la evolución teórica y conceptual del problema de minimización de funciones del cálculo multivariable, lo que nos ha llevado a la formulación del problema de minimización para el aprendizaje profundo en redes neuronales y su resolución mediante el cálculo variacional y la teoría del control.

Para ello, se han utilizado distintas fuentes bibliográficas que difieren no sólo en el nivel de análisis y alcance de la teoría, sino también en la notación utilizada, típica del contexto y problema enfrentado. En esta evolución están implicadas distintas áreas de conocimiento, desde la matemática aplicada, la física y la informática hasta la visión y la inteligencia artificial.

Hemos optado así por mantener las notaciones originales de los artículos y libros consultados, lo que permite entender rápidamente el contexto o "lupa" bajo la cual se está considerando la formulación y el análisis del problema. Al tiempo, hemos intentado relacionar a lo largo de toda la memoria las distintas formulaciones de los problemas para mostrar los pasos y conceptos fundamentales que se repiten, independientemente de la formulación utilizada, en las fuentes consideradas.

1

Optimización con restricciones: el caso finito dimensional

1.1. Introducción

Empezaremos introduciendo, en las secciones 1.2 y 1.3, la notación, los conceptos y las defincciones de optimización con restricciones para problemas finito dimensionales. Esto nos permitirá tender un puente entre la teoría de la optimización sin restricciones vista a lo largo de la carrera y el marco de trabajo de las redes neuronales, cuyo entrenamiento y aprendizaje conduce a considerar y resolver problemas de optimización con restricciones. Pasaremos luego a considerar, en la sección 1.4, la técnica de los multiplicadores de Lagrange. Se trata de una herramienta de resolución de problemas de optimización con restricciones de igualdad, ya que nos proporciona condiciones necesarias de I orden para la determinación de los puntos críticos de una función. También presentaremos, en el teorema 1.4.2, condiciones suficientes de II orden para la clasificación de los puntos críticos a través la determinación, analítica o numérica, de puntos extremos relativos condicionados del campo escalar. En el ejemplo 1.4.1 mostraremos en detalle la aplicación de la técnica a un problema modelo. En la sección siguiente, 1.5, generalizaremos el marco de aplicacación de los multiplicadores de Lagrange considerando restricciones del tipo de igualdad y desigualdad e introduciendo las condiciones necesarias de optimalidad con restricciones de I orden de Karush-Kuhn-Tucker en el teorema 1.5.2. Caracterizaremos nuevamente las condiciones suficientes de II orden en el teorema 1.5.5 y mostraremos su aplicación a un problema de optimización con restricciones de desigualdad en el ejemplo 1.5.1. El material propuesto y ejemplificado en este capítulo constituye la base teórica para la optimización con restricciones finito dimensional.

Optimización con restricciones

La optimización con restricciones puede definirse, de forma general, como la minimización de una función escalar $f : \mathbb{R}^n \to \mathbb{R}$ sujeta a restricciones en las variables que delimitan el conjunto de búsqueda a una región *admisible* del espacio $\Omega \subset \mathbb{R}^n$.

Una posible formulación del problema general de optimización con restricciones es:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad sujeto\, a \begin{cases} g_i(\mathbf{x}) = 0,\ i = 1...p \equiv \mathcal{E} \\ h_i(\mathbf{x}) \leq 0,\ i = 1...m \equiv \mathcal{I} \end{cases} \tag{1.1}$$

donde $\mathbf{x} \in \mathbb{R}^n$ es la variable a optimizar, un vector en general, y la función $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ es la función objetivo o función de coste considerada en el conjunto admisible. Las igualdades:

$$g_i,\ i \in \mathcal{E}$$

son las p **restricciones de igualdad** y las desigualdades:

$$h_i,\ i \in \mathcal{I}$$

son las m **restricciones de desigualdad** que definen el conjunto de soluciones (puntos) admisibles $\Omega \subset \mathbb{R}^n$.

Si no existiesen restricciones ($p = m = 0$), estaríamos ante un problema de minimización sin restricciones, lo que es materia de la asignatura de Cálculo. Se ha añadido un apéndice para recordar los resultados básicos sobre la optimización sin restricciones (apéndice A).

Los problemas de mínimos cuadrados y de programación lineal y cuadrática son casos especiales del problema general de optimización. Se pueden encontrar detalles de las formulaciones correspondientes en el libro de S. Boyd y L. Vandenberghe, *Convex Optimization* [6].

Definición 1.1.1 Definimos el **dominio** $\mathcal{D}$ del problema de minimización con restricciones como el conjunto de puntos donde la función objetivo y todas las funciones de restricciones están bien definidas:

$$\mathcal{D} \doteq \left(\bigcap_{i\in\mathcal{E}} dom\, g_i\right) \cap \left(\bigcap_{i\in\mathcal{I}} dom\, h_i\right)$$

No todos los puntos del dominio $\mathcal{D}$ son admisibles.

Definición 1.1.2 Un punto del dominio, $\mathbf{x} \in \mathcal{D}$ es **admisible** si satisface todas las restricciones. El problema (1.1) es admisible si existe al menos un punto admisible. En caso contrario, el problema se define como no admisible.

El conjunto de todos los puntos admisibles se denomina **conjunto admisible** o conjunto de restricciones, y se define por:

$$\Omega = \{\, \mathbf{x} \in \mathcal{D} \mid g_i(\mathbf{x}) = 0, \quad i \in \mathcal{E}; \quad h_i(\mathbf{x}) \leq 0, \quad i \in \mathcal{I} \,\}$$

Así, podemos escribir el problema (1.1) de manera equivalente, pero más compacta, como sigue:

$$\min_{\mathbf{x}\in\Omega} f(\mathbf{x}) \tag{1.2}$$

El óptimo del problema con restricciones se tendrá que buscar en el conjunto admisible.

Definición 1.1.3 El **valor óptimo** p^* del problema 1.1 se define por:

$$p^* \doteq \text{ínf}\,\{f(\mathbf{x}) \mid g_i(\mathbf{x}) = 0, \quad i = 1...p, \quad h_i(\mathbf{x}) \leq 0, \quad i = 1...m\} \in \mathbb{R}$$

siendo el **punto óptimo**

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\Omega} f(\mathbf{x}) \in \mathbb{R}^n$$

el punto (vector) donde se alcanza el valor óptimo, **mínimo absoluto** de la función en el conjunto de puntos admisibles Ω. En caso de querer maximizar la función definimos

$$p^* \doteq \sup\{f(\mathbf{x}) \mid g_i(\mathbf{x}) = 0, \quad i = 1...p, \quad h_i(\mathbf{x}) \leq 0, \quad i = 1...m\} \in \mathbb{R}$$

siendo el **punto óptimo**

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\Omega} f(\mathbf{x}) \in \mathbb{R}^n$$

el punto (vector) donde se alcanza el valor óptimo, **máximo absoluto** de la función en el conjunto de Ω. Diremos por tanto que $\mathbf{x}^*$ es un **punto óptimo** si $\mathbf{x}^*$ es admisible y $f(\mathbf{x}^*) = p^*$.

1.2. Soluciones locales y globales

Los puntos óptimos pueden ser globales o locales. Se permite que p^* tome los valores $\pm\infty$. Si el problema no es admisible, tendremos $p^* = \infty$ (siguiendo la convención estándar de que el ínfimo de un conjunto vacío es ∞). Si existen puntos admisibles $\mathbf{x}_k$ tales que $f(\mathbf{x}_k) \to -\infty$ cuando $k \to \infty$ entonces $p^* = -\infty$ y diremos que el problema (1.1) no es inferiormente acotado.

Como puede verse en el apéndice A, las soluciones globales son difíciles de encontrar incluso cuando no hay restricciones. La situación puede mejorar cuando añadimos restricciones, ya que el conjunto admisible puede excluir varios mínimos locales y puede ser más sencillo elegir el mínimo global entre los que quedan. Sin embargo, las restricciones también pueden dificultar bastante los cálculos. Un claro ejemplo lo tenemos en el apéndice B, ejemplo B.0.1, en donde una restricción causa la aparición de infinitos mínimos globales.

Definición 1.2.1 Un punto $\mathbf{x}^*$ es una **solución local** del problema 1.2 si $\mathbf{x}^* \in \Omega$ y hay un entorno $\mathcal{N}$ de $\mathbf{x}^*$ tal que $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ para $\mathbf{x} \in \mathcal{N} \cap \Omega$.

Definición 1.2.2 Un punto $\mathbf{x}^*$ es una **solución local estricta** (o solución local fuerte) del problema 1.2 si $\mathbf{x}^* \in \Omega$ y hay un entorno $\mathcal{N}$ de $\mathbf{x}^*$ tal que $f(\mathbf{x}) > f(\mathbf{x}^*)$

para todo $\mathbf{x} \in \mathcal{N} \cap \Omega$ con $\mathbf{x} \neq \mathbf{x}^*$.

Definición 1.2.3 Un punto $\mathbf{x}^*$ es una **solución local aislada** del problema 1.2 si $\mathbf{x}^* \in \Omega$ y hay un entorno $\mathcal{N}$ de $\mathbf{x}^*$ tal que $\mathbf{x}^*$ es la única solución local en $\mathcal{N} \cap \Omega$.

1.3. Regularidad

La regularidad de las funciones objetivo y sus restricciones son factores importantes a la hora de caracterizar las soluciones del problema y elegir las técnicas de optimización. Asegura que la función objetivo y sus restricciones se comportan de una manera razonablemente predecible y, por tanto, permite que los algoritmos funcionen de forma robusta (estable) y controlada (en el error), porque pueden hacer buenas predicciones.

A veces se pueden reformular los problemas de optimización no regulares sin restricciones para transformarlos en problemas de optimización regulares con restricciones. Ilustramos la técnica, conocida con el nombre de **formulación del epígrafe** (Apuntes de fundamentos matemáticos y *Convex Optimization* [5, 6]), considerando el ejemplo B.0.2 1D de minimización sin restricciones no regular, en donde la función objetivo es continua pero no es diferenciable y, por tanto, no hay la regularidad suficiente para imponer la condición necesaria de anulación de la derivada.

La técnica de reformulación utilizada en B.0.2 se denomina **formulación del epígrafe** y es utilizada, entre otros, en casos en los que f es el máximo de una colección de funciones o cuando f es una 1-norma o una ∞-norma de una función vectorial. Gracias a esta formulación se puede afirmar que un objetivo lineal es universal en optimización convexa, ya que cualquier problema se puede reducir a otro con objetivo lineal con restricciones. Se pueden encontrar ejemplos y resultados en el libro de S. Boyd y L. Vandenberghe, *Convex Optimization* [6].

La solución del problema B.0.2, $x = 0$, pertenece a la frontera del conjunto admisible, lo que típicamente se define como conjunto activo.

Definición 1.3.1 El **conjunto activo** $\mathcal{A}(\mathbf{x})$ en cualquier punto admisible $\mathbf{x}$ consiste en los índices de las restricciones de igualdad de $\mathcal{E}$ junto con los índices de las restricciones de desigualdad i para los cuales $h_i(\mathbf{x}) = 0$; es decir:

$$\mathcal{A}(\mathbf{x}) = \mathcal{E} \cup \{\, i \in \mathcal{I} \mid h_i(\mathbf{x}) = 0 \,\}$$

En un punto admisible $\mathbf{x}$, la restricción de desigualdad $i \in \mathcal{I}$ se denomina **activa** si $h_i(\mathbf{x}) = 0$ e **inactiva** si se cumple la desigualdad estricta $h_i(\mathbf{x}) < 0$.

1.4. Multiplicadores de Lagrange

El método de los multiplicadores de Lagrange se utiliza para encontrar el máximo y el mínimo de una función con restricciones de igualdad. Para facilitar la comprensión

de la teoría posterior que utilizaremos, hay que comprender en qué consiste este método. Nos hemos apoyado teóricamente en los libros Cálculo infinitesimal de varias variables de Juan de Burgos [7] y *Multivariate Calculus and Geometry* de Seán Dineen [8].

Las restricciones se imponen con los multiplicadores de Lagrange si la restricción es pertenecer a una curva. Son constantes, ya que no nos encontramos frente a una restricción dinámica.

Sean $f : \mathcal{C} \to \mathbb{R}$ y $\mathbf{g} : \mathcal{C} \to \mathbb{R}^q$ dos funciones de clase C^2 en un abierto $\mathcal{C} \subset \mathbb{R}^p$, siendo $q < p$, tal que ran $d\mathbf{g}(\mathbf{x}) = q$ para $\mathbf{x} \in \mathcal{C}$ y en donde se denota por ran $d\mathbf{g}(\mathbf{x})$ al rango de la matriz jacobiana de la función vectorial $\mathbf{g}$. Se quieren hallar los extremos relativos de f condicionados por la restricción vectorial $\mathbf{g}(\mathbf{x}) = \mathbf{0}$. Para ello, construimos la función de Lagrange

$$L(\mathbf{x}) = f(\mathbf{x}) + \lambda_1 g_1(\mathbf{x}) + \cdots + \lambda_q g_q(\mathbf{x})$$

donde $\lambda_1, \cdots, \lambda_q \in \mathbb{R}$ son los denominados multiplicadores de Lagrange; y siendo $g_1, \cdots, g_q$ las componentes de $\mathbf{g} = (g_i)_{i=1}^q$.

Definición 1.4.1 Recordamos que se llaman **puntos críticos** de la función diferenciable f a aquellos $\mathbf{x} \in \mathcal{C}$ tales que $df(\mathbf{x}) = \mathbf{0}$.

Para encontrar los extremos de una función con restricciones utilizando el método de los multiplicadores de Lagrange (obtenido de Cálculo infinitesimal de varias variables [7]), procedemos del siguiente modo:

1. **Condiciones necesarias de I orden**

Se resuelve el sistema de ecuaciones $dL(\mathbf{x}) = \mathbf{0}$ (p ecuaciones) y $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ (q ecuaciones), es decir, se determinan los $\mathbf{x} = (x_1, ..., x_p) \in \mathcal{C}$ y los $\boldsymbol{\lambda}_\mathbf{x} = (\lambda_1, ..., \lambda_q) \in \mathbb{R}^q$ para los que

$$\begin{gathered} \frac{\partial f(\mathbf{x})}{\partial x_i} + \lambda_1 \frac{\partial g_1(\mathbf{x})}{\partial x_i} + \ldots + \lambda_q \frac{\partial g_q(\mathbf{x})}{\partial x_i} = 0 \quad (i = 1, ..., p) \\ \text{y} \quad g_j(\mathbf{x}) = 0 \quad (j = 1, ..., q) \end{gathered} \tag{1.3}$$

Para que en un punto $\mathbf{x} = \mathbf{a}$ haya extremo condicionado es necesario que $\mathbf{x} = \mathbf{a}$ y $\boldsymbol{\lambda}_\mathbf{x} = \boldsymbol{\lambda}_\mathbf{a}$ sean soluciones del sistema (1.3).

2. **Condiciones suficientes de II orden**

Para analizar si la función f tiene un extremo condicionado por $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ en un punto $\mathbf{a}$ (solución de (1.3)), se determinan el espacio vectorial

$$V(\mathbf{a}) = \{\Delta\mathbf{x} \in \mathbb{R}^p / d\mathbf{g}(\mathbf{a})(\Delta\mathbf{x}) = 0\}$$

de las direcciones ortogonales al gradiente y por tanto tangentes a las curvas de nivel de g_i, la diferencial segunda $d^2 g(\mathbf{a})$ (matriz hessiana de la función lagrangiana) cuya

actuación está definida por la forma cuadrática (para $p = 2$ y con $\Delta\mathbf{x} = (dx, dy)$)

$$d^2L(\mathbf{a})(\Delta\mathbf{x}) = d^2L(\mathbf{a})(dx, dy) = (\Delta\mathbf{x})' d^2L(\mathbf{a})\Delta\mathbf{x} =$$

$$= L_{xx}(\mathbf{a})dx^2 + 2L_{xy}(\mathbf{a})dxdy + L_{yy}(\mathbf{a})dy^2$$

y la restricción $\Omega = d^2L(\mathbf{a})|_V$, también cuadrática, que se define resolviendo la ecuación (condición de pertenecer a la curva restricción) $d\mathbf{g}(\mathbf{a})(\Delta\mathbf{x}) = 0$, es decir determinando $\Delta\mathbf{x} \in V(\mathbf{a})$.

Apoyándonos en la definición y clasificación de formas cuadráticas (definición A.2.4 en el anexo), se tiene el siguiente criterio de clasificación de puntos críticos:

Criterio de clasificación de puntos críticos

Teorema 1.4.2 *Sea $\Omega = d^2L(\mathbf{a})|_V$ la forma cuadrática restringida a $V(\mathbf{a})$ asociada a un punto crítico del lagrangiano aumentado L del problema de minimización con restricciones.*

1. *Si Ω es definida positiva, entonces f tiene en $\mathbf{a}$ un mínimo relativo condicionado por la ligadura $\mathbf{g}(\mathbf{x}) = \mathbf{0}$.*
2. *Si Ω es definida negativa, entonces f tiene en $\mathbf{a}$ un máximo relativo condicionado por la ligadura $\mathbf{g}(\mathbf{x}) = \mathbf{0}$.*
3. *Si Ω es semidefinida, este método no informa sobre si hay un extremo en $\mathbf{a}$.*
4. *Si Ω no es definida ni semidefinida, entonces f no tiene en $\mathbf{a}$ un extremo relativo condicionado por la ligadura $\mathbf{g}(\mathbf{x}) = \mathbf{0}$.*

Para ilustrar la aplicación de esta teoría a un problema de optimización con restricción consideramos el siguiente ejemplo, propuesto en el libro de Burgos, Cálculo infinitesimal de varias variables [7], suficientemente simple para poder realizar los cálculos de forma analítica. En el caso general, la resolución del sistema para el cálculo de los multiplicadores tiene que ser numérica mediante algoritmos.

Ejemplo 1.4.1 Sean $f(x, y)$, $g(x, y)$ campos escalares definidos por

$$f(x, y) = xy \qquad g(x, y) = x^2 + y^2 + xy - 4$$

Consideramos los problemas de optimización

$$\text{máx}\, f(x, y) \text{ y } \text{mín}\, f(x, y) \qquad \text{s.a} \quad g(x, y) = 0$$

Buscamos hallar los valores máximo y mínimo que alcanza $f(x, y)$ cuando (x, y) recorre la elipse $x^2 + y^2 + xy = 4$.

Como f es continua en $\mathbb{R}^2$ y la elipse es un conjunto compacto[1] de $\mathbb{R}^2$, podemos afirmar por el teorema de Weierstrass que existen los extremos pedidos.

[1] Cerrado y acotado.

Vamos a aplicar el método de los multiplicadores de Lagrange para el problema de hallar los extremos absolutos de $f(x,y)$ a lo largo de la curva de nivel $g(x,y)=0$, es decir:

$$\text{Opt } f(x,y), \qquad \text{s.a.} \qquad g(x,y)=x^2+y^2+xy-4=0$$

En el marco general se tiene $p=2$ (variables) y $q=1$ (restricciones) luego $q<p$ y existen infinitos puntos admisibles dados por todos los puntos de la elipse. La matriz jacobiana solo tiene una fila dada por el gradiente de f. Definiendo

$$\Omega=\left\{(x,y)\in\mathbb{R}^2/g(x,y)=0\right\}$$

el problema de optimización con restricciones consiste en calcular

$$\text{Opt}_{(x,y)\in\Omega} f(x,y)$$

Para ello empezamos determinando todos los puntos estacionarios de la función de Lagrange que verifican la restricción. Luego pasaremos a su clasificación[2].

Por tanto, aplicamos el método de los multiplicadores de Lagrange, para lo que construimos la función (lagrangiano aumentado)

$$L(x,y,\lambda)=f(x,y)+\lambda g(x,y)=xy+\lambda\left(x^2+y^2+xy-4\right)$$

y consideramos el sistema no lineal de $p+q=3$ ecuaciones

$$\begin{cases} L_x(x,y,\lambda) &= y+\lambda(2x+y)=0 \\ L_y(x,y,\lambda) &= x+\lambda(2y+x)=0 \\ g(x,y) &= x^2+y^2+xy-4=0 \end{cases}$$

definido por la condición necesaria de primer orden dada por la anulación del gradiente $\nabla L(x,y,\lambda)$ del lagrangiano aumentado en el conjunto de puntos admisibles, que es donde se verifica la restricción.

El sistema $\nabla L=(0,0,0)'$ es no lineal (ya que es cuadrático) y acoplado. Se puede desacoplar reconduciéndose a una única ecuación cuadrática para x o y y para cada valor de λ. Para ello, factorizamos en la primera y segunda ecuación del sistema el término

$$\lambda(x+y)$$

Igualando se deduce

$$y+\lambda x=x+\lambda y$$

es decir

$$y-x=\lambda(y-x) \tag{1.4}$$

[2]Observamos que el teorema 1.4.2 proporciona condiciones suficientes para extremos condicionados relativos. Los puntos pedidos, extremos absolutos, serán aquellos de todos los puntos extremos condicionados relativos en los que f alcanza los valores extremos.

luego $\lambda = 1$ si $x \neq y$.

Determinado el primer multiplicador, sustituimos en las primeras dos ecuaciones del sistema para obtener $x = -y \neq 0$.

Sustituyendo en la tercera ecuación tenemos $g(x, -x) = x^2 - 4$, de donde $x = \pm 2$.

La ecuación (1.4) se verifica también si $x = y \neq 0$. en cuyo caso $\lambda = -1/3$. Sustituyendo $g(x, x) = 3x^2 - 4$ de donde $x = \pm 2/\sqrt{3}$.

Las soluciones de este sistema son las siguientes:

$$\mathbf{a}_1 = (x_1, y_1; \lambda_1) = (2/\sqrt{3}, 2/\sqrt{3}; -1/3) \quad \mathbf{a}_2 = (x_2, y_2; \lambda_2) = (-2/\sqrt{3}, -2/\sqrt{3}; -1/3)$$

$$\mathbf{a}_3 = (x_3, y_3; \lambda_3) = (2, -2; 1) \qquad \mathbf{a}_4 = (x_4, y_4; \lambda_4) = (-2, 2; 1)$$

Para su clasificación vamos a determinar el espacio vectorial de dimensión $p = 2$

$$V(\mathbf{a}) = \{\Delta\mathbf{x} \in \mathbb{R}^p / dg(\mathbf{a})(\Delta\mathbf{x}) = \mathbf{0}\}$$

Puesto que g es diferenciable, existe una aplicación lineal (su diferencial $dg(\mathbf{a})$) tal que

$$dg(\mathbf{a})(\Delta\mathbf{x}) = \nabla g(\mathbf{a}) \cdot \Delta\mathbf{x} = 0$$

y redefinimos

$$V(\mathbf{a}) = \{\Delta\mathbf{x} \in \mathbb{R}^p / dg(\mathbf{a})(\Delta\mathbf{x}) = \nabla g(\mathbf{a}) \cdot \Delta\mathbf{x} = 0\}$$

Para construir la aplicación calculamos el gradiente

$$\nabla g(\mathbf{a}) = (2x + y, 2y + x)$$

y evaluamos en los 4 puntos estacionarios (críticos) para obtener

$$\begin{aligned}
\nabla g(\mathbf{a}_3) = \nabla g(2, -2, 1) = (2, -2),\\
\nabla g(\mathbf{a}_4) = \nabla g(-2, 2, 1) = (-2, 2),\\
\nabla g(\mathbf{a}_1) = \nabla g(2\sqrt{3}, 2\sqrt{3}, -1/3) = (6\sqrt{3}, 6\sqrt{3}),\\
\nabla g(\mathbf{a}_2) = \nabla g(-2\sqrt{3}, -2\sqrt{3}, -1/3) = (-6\sqrt{3}, -6\sqrt{3}))
\end{aligned}$$

luego definimos $\Delta\mathbf{x} = (dx, dy)$ e imponemos la condición de pertenecer a la curva dada por $\nabla g(\mathbf{a}) \cdot \Delta\mathbf{x} = 0$ siendo $\mathbf{a}$ un punto crítico. Se tiene

$$\nabla g(2, -2, 1) \cdot (dx, dy) = (2, -2) \cdot (dx, dy) = 2(dx - dy) = 0$$

luego la condición se verifica para $dx = dy$ y se deduce

$$V(\mathbf{a}_1) = \left\{\Delta\mathbf{x} = (dx, dy) \in \mathbb{R}^2 / \Delta\mathbf{x} = (dx, dx), dx \neq 0.\right\}$$

Observamos que el origen $(0, 0)$ no pertenece a la elipse, ya que no verifica la res-

tricción $g(x, y) = 0$ y no es un punto admisible. Esto implica $dx = dy \neq 0$ en la definición de $V(\mathbf{a}_1)$.

Pasando al segundo punto

$$\nabla g(-2, 2, 1) \cdot (dx, dy) = (-2, 2) \cdot (dx, dy) = -2(dx - dy) = 0,$$

y nuevamente la condición se verifica para $dx = dy$ luego $V(\mathbf{a}_2) = V(\mathbf{a}_1)$.

En el tercer punto se tiene

$$\nabla g(2\sqrt{3}, 2\sqrt{3}, -1/3) \cdot (dx, dy) = (6\sqrt{3}, 6\sqrt{3}) \cdot (dx, dy) = 6\sqrt{3}(dx + dy) = 0$$

la condición se verifica para $dx = -dy$, luego

$$V(\mathbf{a}_3) = \left\{ \Delta\mathbf{x} = (dx, dy) \in \mathbb{R}^2 \,/\, \Delta\mathbf{x} = (dx, -dx),\ \ dx \neq 0 \right\}$$

En el cuarto punto

$$\nabla g(-2\sqrt{3}, -2\sqrt{3}, -1/3) \cdot (dx, dy) = (-6\sqrt{3}, -6\sqrt{3})) \cdot (dx, dy) = -6\sqrt{3}(dx + dy)$$

la condición se verifica para $dx = -dy$ luego $V(\mathbf{a}_3) = V(\mathbf{a}_4)$.

La determinación del espacio de las direcciones tangentes a la curva restricción permitirá la evaluación de la forma cuadrática en los puntos admisibles del problema. Para determinar la forma cuadrática $d^2L(\mathbf{a})(\Delta\mathbf{x})$ calculamos la matriz hessiana del lagrangiano

$$L_{xx}(x, y) = 2\lambda, \qquad L_{xy}(x, y) = L_{yx}(x, y) = 1 + \lambda, \qquad L_{yy}(x, y) = 2\lambda$$

de donde la matriz hessiana es, en términos del multiplicador,

$$H(\lambda) = \begin{pmatrix} 2\lambda & 1 + \lambda \\ 1 + \lambda & 2\lambda \end{pmatrix}$$

Evaluando los valores de los multiplicadores calculados antes

$$H(1) = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}, \qquad H(-1/3) = \begin{pmatrix} -2/3 & 2/3 \\ 2/3 & -2/3 \end{pmatrix}$$

Clasificación: Para clasificar los puntos, calculamos la forma cuadrática

$$d^2L(\mathbf{a})(\Delta\mathbf{x}) = \begin{pmatrix} dx & dy \end{pmatrix} \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = 2dx^2 + 4dxdy + 2dy^2$$

y usamos $V(\mathbf{a}) = (dx, dx)$ para obtener

$$d^2L(\mathbf{a})(\Delta\mathbf{x})|_{V'} = \left(2dx^2 + 4dxdy + 2dy^2\right)|_V = 8dx^2 > 0$$

para $dx = dy \neq 0$. Por el teorema (1.4.2) el campo f tiene en $\mathbf{a}$ un mínimo relativo condicionado por la ligadura $g(\mathbf{x}) = 0$.

Definimos ahora la forma cuadrática

$$d^2L(\mathbf{a})(\Delta\mathbf{x}) = \begin{pmatrix} dx & dy \end{pmatrix} \begin{pmatrix} -2/3 & 2/3 \\ 2/3 & -2/3 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} =$$

$$= -\frac{2}{3}dx^2 + \frac{4}{3}dxdy - \frac{2}{3}dy^2$$

para $dx = -dy \neq 0$.

Para $V(\mathbf{a}) = (dx, -dx)$ obtenemos

$$d^2L(\mathbf{a})(\Delta\mathbf{x})|_{V'} = \left(-\frac{2}{3}dx^2 + \frac{4}{3}dxdy - \frac{2}{3}dy^2\right)|_{V'} = -\frac{8}{3}dx^2 < 0$$

para $dx = -dy \neq 0$. Por el teorema 1.4.2, el campo f tiene en $\mathbf{a}$ un máximo relativo condicionado por la ligadura $g(\mathbf{x}) = 0$.

Los valores de f en los cuatro puntos estacionarios anteriores (que son los únicos por la diferenciabilidad de L en todo el plano) son:

$$f(x_1, y_1) = 4/3, \quad f(x_2, y_2) = 4/3, \quad f(x_3, y_3) = -4, \quad f(x_4, y_4) = -4$$

por lo que los valores pedidos son:

máximo: $4/3$, que se alcanza a lo largo de la recta $x = y$ y cumple la restricción en $(2/\sqrt{3}, 2/\sqrt{3})$ y en $(-2/\sqrt{3}, -2/\sqrt{3})$

mínimo: -4, que se alcanza a lo largo de la recta $x = y$ y cumple la restricción en $(2, -2)$ y en $(-2, 2)$

En la figura 1.1 se muestra una representacción gráfica de los resultados obtenidos:

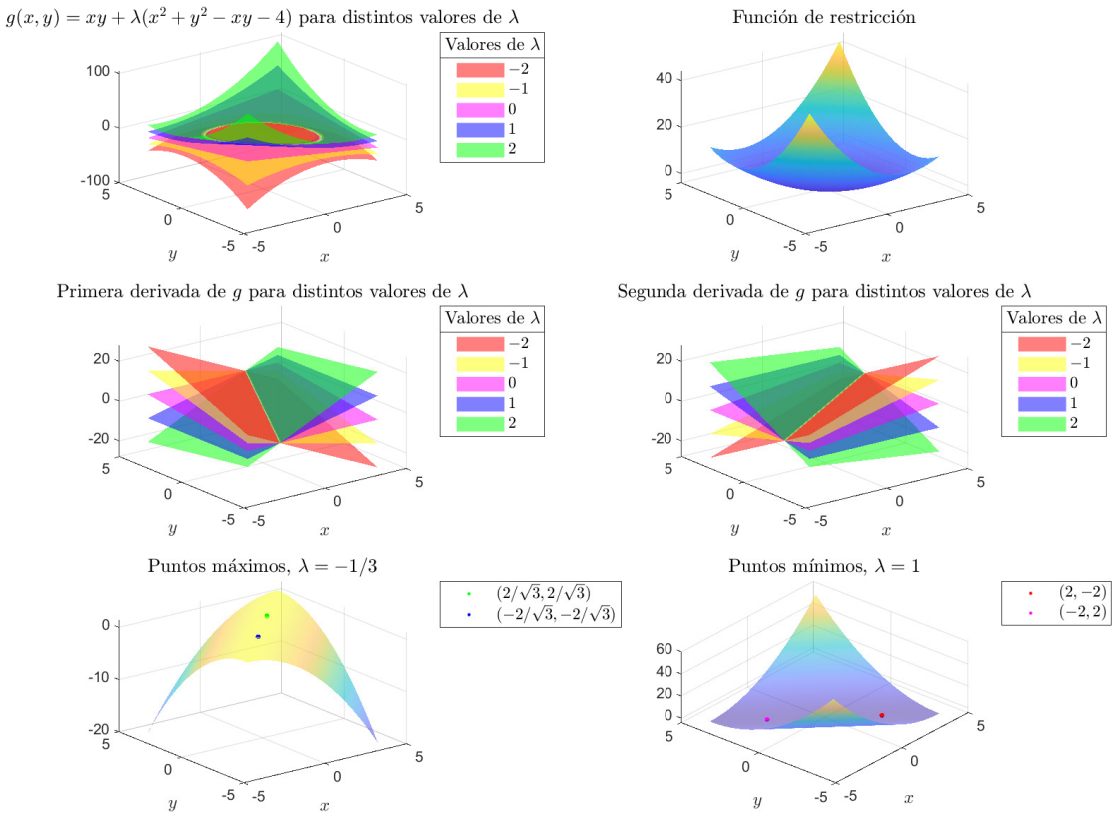

Figura 1.1: Ilustración del ejemplo de multiplicadores de Lagrange. Primera fila: a la izquierda, la familia paramétrica $g(x, y, \lambda)$ para distintos valores de λ. A la derecha, la restricción de pertenecer a la elipse de ecuación $g = 0$. Segunda fila: se representan las componentes del gradiente de L para distintos valores paramétricos. Tercera fila: Representación de los valores extremos para los multiplicadores de Lagrange calculados. A la izquierda, la gráfica de la función cóncava $L(x, y, -1/3)$ y a la derecha, la gráfica de la función convexa $L(x, y, 1)$.

1.5. Condiciones de Karush-Kuhn-Tucker

Las condiciones de Karush-Kuhn-Tucker (KKT) (*Numerical Optimization*, *Convex Optimization* [4, 6]) representan una generalización del método de los multiplicadores de Lagrange al permitir introducir en la formulación del problema de optimización un conjunto de restricciones de desigualdad, como podemos leer en El teorema de Karush-Kuhn-Tucker, una generalización del teorema de los multiplicadores de Lagrange, y programación convexa de Francisco Javier Martínez Sánchez [9]. Representan la base de lo que se conoce como programación no lineal, y en particular de la programación convexa, la cuadrática y la lineal. La discretización de estas condiciones y su resolución numérica representan la base de los algoritmos de optimización con restricciones.

Como en el caso sin restricciones (apéndice A) y en el caso con restricciones de igualdad tratado en la sección anterior, vamos a discutir condiciones de optimalidad de dos tipos. Las condiciones *necesarias* son aquellas que debe cumplir cualquier punto crítico, candidato a ser solución del problema de optimización. Las condiciones *suficientes* son aquellas que, de ser satisfechas por un punto concreto $\mathbf{x}^*$, garantizan que $\mathbf{x}^*$ es un extremos relativo, es decir una solución del problema.

Descubriremos nuevamente que las condiciones suficientes se basan en hipótesis de convexidad de la función objetivo y del conjunto admisible definido por las funciones restricciones. La definición de convexidad se encuentra en el apéndice A.

En el camino hacia una formulación continua de una red neuronal basada en EDOs observamos que, a pesar de la generalidad alcanzada por el marco de las condiciones KKT (que no entran en el currículo del grado de Matemáticas), estamos todavía optimizando en espacios de dimensión finita y las soluciones buscadas son puntos de algún espacio vectorial $\mathbb{R}^n$. Cuando pasemos a considerar restricciones basadas en EDOs no lineales necesitaremos generalizar aún más el marco teórico para llegar a la minimización de funcionales cuyos *puntos* son funciones en espacios infinito dimensionales. Para ello necesitaremos el marco del cálculo variacional.

1.5.1. Condiciones de optimalidad de primer orden

Para definir estas condiciones, tenemos que definir la función lagrangiana asociada al problema general de minimización con restricciones (1.1), cuya formulación recordamos aquí para conveniencia del lector:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad s.a \quad \begin{cases} g_i(\mathbf{x}) = 0,\ i = 1...p \equiv \mathcal{E} \\ h_i(\mathbf{x}) \leq 0,\ i = 1...m \equiv \mathcal{I} \end{cases}$$

La lagrangiana del problema de minimización es una función (campo escalar):

$$\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$$

definida por:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) = f_0(\mathbf{x}) + \sum_{i=1}^{m} \nu_i h_i(\mathbf{x}) + \sum_{i=1}^{p} \lambda_i g_i(\mathbf{x}) \tag{1.5}$$

donde los $\nu_i \geq 0$ son los multiplicadores asociados a las restricciones de desigualdad; los λ_i son los multiplicadores asociados a las restricciones de igualdad, y los vectores $\boldsymbol{\nu} \in \mathbb{R}^m$ y $\boldsymbol{\lambda} \in \mathbb{R}^p$ son las **variables duales del problema**.

La formulación de la lagrangiana asociada al problema de minimización con restricciones (1.1) permite recconducir el problema a uno de minimización sin restricciones. Las restricciones pasan a aparecen incluidas dentro de la función a minimizar, en términos de penalizaciones definidas mediante multiplicadores.

Se ha añadido en el anexo C un apartado con todos los conceptos y resultados necesarios para facilitar la comprensión y posterior demostración de las condiciones de optimalidad. Vamos a empezar con una definición que nos servirá de hipótesis en muchos casos.

Definición 1.5.1 Dado el punto $\mathbf{x}$ y el conjunto activo $\mathcal{A}(\mathbf{x})$, decimos que la **cualificación de restricción de independencia lineal (CRIL)** se verifica si el conjunto de gradientes de restricciones activas $\nabla c_i(\mathbf{x}), i \in \mathcal{A}(\mathbf{x})$ es linealmente independiente. En general, si la CRIL se verifica, ninguno de los gradientes puede ser cero.

Las siguientes condiciones se denominan de primer orden porque están relacionadas con las propiedades de los gradientes (vectores de la primera derivada) de la función objetivo y las restricciones.

Teorema 1.5.2 ***Condiciones necesarias de primer orden***
Sean $\mathbf{x}^*$ solución local de (1.1), y las funciones f, g_i y h_i en (1.1) continuamente diferenciables. Consideramos también que la CRIL se verifica en $\mathbf{x}^*$.

Entonces existe un vector de multiplicadores de Lagrange ν_i^*, $i \in \mathcal{E} \cup \mathcal{I}$ tal que las siguientes condiciones se satisfacen en $(\mathbf{x}^*, \boldsymbol{\nu}^*)$:

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) = 0 \tag{1.6a}$$

$$g_i(\mathbf{x}^*) = 0, \quad \forall i \in \mathcal{E} \tag{1.6b}$$

$$h_i(\mathbf{x}^*) \leq 0, \quad \forall i \in \mathcal{I} \tag{1.6c}$$

$$\nu_i^* \geq 0, \quad \forall i \in \mathcal{I} \tag{1.6d}$$

$$\nu_i^* h_i(\mathbf{x}) = 0, \quad \forall i \in \mathcal{I} \tag{1.6e}$$

Las condiciones (1.6) se denominan **condiciones de Karush-Kuhn-Tucker**, o condiciones KKT para abreviar. La condición (1.6e) es de complementariedad; implica que, o bien la restricción i es activa, o bien $\nu_i^* = 0$, o posiblemente ambas; es decir, $\nu_i^* \in \mathcal{A}(\mathbf{x})$.

1.5.2. Condiciones de segundo orden

Hasta ahora hemos visto que las condiciones de primer orden nos dicen cómo se relacionan entre sí las primeras derivadas de f y las restricciones activas h_i en una solución $\mathbf{x}^*$. Cuando se satisfacen estas condiciones, un movimiento a lo largo de un vector $\mathbf{w}$ de $\mathcal{F}(\mathbf{x}^*)$ incrementa la aproximación de primer orden a la función objetivo (es decir, $\mathbf{w}^T \nabla f(\mathbf{x}^*) < 0$), o mantiene su valor ($\mathbf{w}^T \nabla f(\mathbf{x}^*) = 0$).

Las segundas derivadas de f y las restricciones h_i se encargan de "romper empates" en cuanto a condiciones de optimalidad. Para las direcciones $\mathbf{w} \in \mathcal{F}(\mathbf{x}^*) = 0$, si solo conocemos información de la primera derivada, no podremos determinar si un movimiento en esa dirección aumentará o disminuirá el valor de la función objetivo. Las condiciones de segundo orden examinan los términos de la segunda derivada de la expansión en serie de Taylor de f, g_i y h_i para comprobar si esta información extra resuelve nuestra duda.

Como vamos a tratar con segundas derivadas, necesitamos suponer que f, g_i y h_i son dos veces continuamente diferenciables.

Definición 1.5.3 Dado $\mathcal{F}(\mathbf{x}^*)$ de la definición C.1.3 y un vector de multiplicadores de Lagrange $\boldsymbol{\nu}^*$ que satisface las condiciones KKT 1.6, definimos el **cono crítico** $\mathcal{C}(\mathbf{x}^*, \boldsymbol{\nu}^*)$ como sigue:

$$\mathcal{C}(\mathbf{x}^*, \boldsymbol{\nu}^*) = \left\{ \mathbf{w} \in \mathcal{F}(\mathbf{x}^*) \mid \nabla c_i(\mathbf{x}^*)^T \mathbf{w} = 0, \forall i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ con } \nu_i^* > 0 \right\}$$

Equivalentemente,

$$\mathbf{w} \in \mathcal{C}\left(\mathbf{x}^*, \boldsymbol{\nu}^*\right) \Longleftrightarrow \begin{cases} \nabla c_i\left(\mathbf{x}^*\right)^T \mathbf{w} = 0, & \forall i \in \mathcal{E} \\ \nabla c_i\left(\mathbf{x}^*\right)^T \mathbf{w} = 0, & \forall i \in \mathcal{A}\left(\mathbf{x}^*\right) \cap \mathcal{I} \text{ con } \nu_i^* > 0 \\ \nabla c_i\left(\mathbf{x}^*\right)^T \mathbf{w} \geq 0, & \forall i \in \mathcal{A}\left(\mathbf{x}^*\right) \cap \mathcal{I} \text{ con } \nu_i^* = 0 \end{cases} \tag{1.7}$$

El cono crítico contiene aquellas direcciones $\mathbf{w}$ que tenderían a "adherirse" a las condiciones de desigualdad incluso si hiciésemos pequeños cambios a la función objetivo y también a las condiciones de igualdad. De la definición anterior y del hecho de que $\nu_i^* = 0$ para todos los componentes inactivos $i \in \mathcal{I} \backslash \mathcal{A}(\mathbf{x}^*)$, tenemos de manera inmediata que

$$\mathbf{w} \in \mathcal{C}(\mathbf{x}^*, \boldsymbol{\nu}^*) \;\Rightarrow\; \nu_i^* \nabla c_i(\mathbf{x}^*)^T \mathbf{w} = 0 \qquad \forall i \in \mathcal{E} \cup \mathcal{I} \tag{1.8}$$

Por tanto, de la primera condición KKT (1.6a) y la definición 1.5, tenemos que

$$\mathbf{w} \in \mathcal{C}(\mathbf{x}^*, \boldsymbol{\nu}^*) \;\Rightarrow\; \mathbf{w}^T \nabla f(\mathbf{x}^*) = \sum_{i \in \mathcal{E} \cup \mathcal{I}} \nu_i^* \mathbf{w}^T \nabla c_i(\mathbf{x}^*) = 0$$

Podemos resumir que el cono crítico $\mathcal{C}(\mathbf{x}^*, \boldsymbol{\nu}^*)$ contiene direcciones de $\mathcal{F}(\mathbf{x}^*)$ para las que la información de la primera derivada es insuficiente para saber si f va a crecer o decrecer.

Teorema 1.5.4 ***Condiciones necesarias de segundo orden***
Supongamos que $\mathbf{x}^*$ *es una solución local de 1.1 y que la CRIL se satisface. Sea* $\boldsymbol{\nu}^*$ *el vector de multiplicadores de Lagrange para el cual se satisfacen las condiciones KKT en (1.6). Entonces:*

$$\mathbf{w}^T \nabla^2_{\mathbf{xx}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\nu}^*)\, \mathbf{w} \geq 0, \qquad \forall \mathbf{w} \in \mathcal{C}(\mathbf{x}^*, \boldsymbol{\nu}^*). \tag{1.9}$$

En el caso de problemas de maximización, la desigualdad cambia de signo, ya que estamos buscando concavidad (no convexidad como en el caso de problemas de minimización).

Las condiciones suficientes son condiciones sobre f, $g_i, i \in \mathcal{E}$; y $h_i, i \in \mathcal{I}$ que nos aseguran que $\mathbf{x}^*$ es solución local del problema (1.1). Las condiciones suficientes son muy parecidas a las necesarias, pero difieren en que la cualificación de restricciones no es necesaria y la desigualdad en (1.9) pasa a ser estricta.

Teorema 1.5.5 ***Condiciones suficientes de segundo orden***
Supongamos que para algún punto admisible $\mathbf{x}^* \in \mathbb{R}^n$ *existe un vector de multiplicadores de Lagrange* $\boldsymbol{\nu}^*$ *tal que las condiciones KKT* (1.6) *se satisfacen. Supongamos también que:*

$$\mathbf{w}^T \nabla^2_{\mathbf{xx}} \mathcal{L}\left(\mathbf{x}^*, \boldsymbol{\nu}^*\right) \mathbf{w} > 0, \qquad \forall\, \mathbf{w} \in \mathcal{C}\left(\mathbf{x}^*, \boldsymbol{\nu}^*\right), \quad \mathbf{w} \neq \mathbf{0}.$$

Entonces, $\mathbf{x}^*$ *es una solución local estricta para* (1.1)*. De nuevo, en el caso de problemas de maximización, la desigualdad cambia de signo.*

Las demostraciones de estos teoremas son largas y de notación bastante pesada, por lo que no vamos a escribirlas. Pueden encontrarse en el capítulo 12 del libro de Nocedal y Wright, *Numerical optimization* [4].

Ejemplo 1.5.1 Sea

$$f(x, y) = xy$$

Consideramos el problema de maximización con restricciones:

$$\text{máx}\, f(x, y) \qquad \text{s.a.} \begin{cases} x + y^2 & \leq 2 \\ x, y & \geq 0 \end{cases}$$

y el punto $\mathbf{x}^* = \left(4/3, \sqrt{2/3}\right)$.

Se pide:

1. Comprobar que se verifican las condiciones necesarias KKT en $\mathbf{x}^*$.
2. Comprobar si se cumplen las condiciones suficientes de segundo orden.

El enunciado de este problema se puede encontrar en *A karush-kuhn-tucker example* de R. B. Israel, profesor de la universidad de Columbia [10].

Apartado 1. En primer lugar, vamos a comprobar si se verifican las condiciones KKT. Podemos reescribir el problema de la siguiente forma:

$$\text{máx}\, xy \qquad \text{s.a.} \begin{cases} r_1 \coloneqq & x + y^2 - 2 & \leq 0 \\ r_2 \coloneqq & -x & \leq 0 \\ r_3 \coloneqq & -y & \leq 0 \end{cases}$$

Vamos a analizar las condiciones KKT. Introducimos los tres multiplicadores de Lagrange correspondientes a las tres restricciones de desigualdad mediante $\boldsymbol{\nu} = (\nu_1, \nu_2, \nu_3)'$. Como no hay restricciones de igualdad, no introducimos multiplicadores del tipo $\boldsymbol{\lambda}$ y definimos $\mathcal{L}(\mathbf{x}, \boldsymbol{\nu}, \boldsymbol{\lambda}) = \mathcal{L}(\mathbf{x}, \boldsymbol{\nu})$.

El lagrangiano aumentado es

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\nu}) = xy + \nu_1(x + y^2 - 2) - x\nu_2 - y\nu_3$$

El gradiente del lagrangiano es

$$\nabla\mathcal{L}(\mathbf{x}, \boldsymbol{\nu}) = \begin{pmatrix} y + \nu_1 - \nu_2 \\ x + 2y\nu_1 - \nu_3 \end{pmatrix}$$

Las condiciones KKT son, por tanto:

$$\begin{aligned} y + \nu_1 - \nu_2 &= 0 \\ x + 2y\nu_1 - \nu_3 &= 0 \\ x + y^2 - 2 &\leq 0 \\ -x, -y &\leq 0 \\ \nu_1, \nu_2, \nu_3 &\leq 0 \\ \nu_1(x + y^2 - 2) &= 0 \\ \nu_2(-x) &= 0 \\ \nu_3(-y) &= 0 \end{aligned}$$

NOTA: Dado que estamos hablando de un problema de maximización, las condiciones (1.6d) cambian de signo.

Sustituyendo $\mathbf{x}^*$ en las condiciones tenemos:

$$\begin{aligned}
\sqrt{2/3} + \nu_1 - \nu_2 &= 0 \\
4/3 + 2\sqrt{2/3}\nu_1 - \nu_3 &= 0 \\
4/3 + (\sqrt{2/3})^2 - 2 &\leq 0 \\
-4/3, -\sqrt{2/3} &\leq 0 \\
\nu_1, \nu_2, \nu_3 &\leq 0 \\
\nu_1(4/3 + (\sqrt{2/3})^2 - 2) &= 0 \\
\nu_2(-4/3) &= 0 \\
\nu_3(-\sqrt{2/3}) &= 0
\end{aligned}$$

Inmediatamente se sigue que $\nu_2 = \nu_3 = 0$.

Sustituyendo estos valores en el sistema anterior, nos queda:

$$\begin{aligned}
\sqrt{2/3} + \nu_1 &= 0 \\
\nu_1 &\leq 0
\end{aligned}$$

Por lo que concluimos que $\nu_1 = -\sqrt{2/3}$.

Así pues, el punto $(4/3, \sqrt{2/3})$ es una solución admisible que solo satura la primera restricción. Como solo hay una restricción activa, la CRIL se verifica para $\mathbf{x}^*$. El gradiente de r_1 es

$$\nabla r_1(\mathbf{x}^*) \;=\; (\,1 \quad 2y\,) \;=\; (\,1 \quad 2\sqrt{2/3}\,)$$

Por tanto, tenemos

$$x = 4/3, \quad y = \sqrt{2/3}\,; \qquad \nu_1 = -\sqrt{2/3}\,, \quad \nu_2, \nu_3 = 0$$

y podemos afirmar que se cumplen las condiciones necesarias KKT.

Apartado 2. Ahora vamos a estudiar si se cumplen las condiciones suficientes de segundo orden.

Ya sabemos que el punto $\mathbf{x}^*$ solo satura la primera restricción. Para definir el cono crítico vamos a utilizar (1.8):

$$\mathbf{w} \in \mathcal{C}(\mathbf{x}^*, \boldsymbol{\nu}^*) \;\Rightarrow\; \nu_i^* \nabla c_i(\mathbf{x}^*)^T \mathbf{w} = 0 \qquad \forall i \in \mathcal{E} \cup \mathcal{I}$$

Queremos calcular $\mathbf{w} = (d_1, d_2)$. Con la primera restricción, la única activa, imponemos

$$\nu_1 \nabla c_1(\mathbf{x}^*)^T \mathbf{w} = \nu_1 \nabla c_1(\mathbf{x}^*)^T \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = 0$$

Sustituyendo los valores calculados anteriormente y operando

$$-\sqrt{2/3}\begin{pmatrix} 1 & 2\sqrt{2/3} \end{pmatrix}\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = -\sqrt{\frac{2}{3}}\,d_1 - \frac{4}{3}d_2 = 0$$

de donde

$$d_2 = -\frac{3}{4}\sqrt{\frac{2}{3}}\,d_1$$

Por tanto, el cono crítico es un espacio vectorial de dimensión 1

$$\mathcal{C}\left(\mathbf{x}^*, \boldsymbol{\lambda}^*\right) = \left\{ \begin{pmatrix} d_1 & -\frac{3}{4}\sqrt{\frac{2}{3}}\,d_1 \end{pmatrix}, \quad d_1 \neq 0 \right\}$$

Pasamos al estudio de la forma cuadrática asociada a la matriz hessiana del lagrangiano

$$\mathbf{w}^T \nabla^2_{\mathbf{xx}} \mathcal{L}\left(\mathbf{x}^*, \boldsymbol{\nu}^*\right) \mathbf{w} = \begin{pmatrix} d_1 & -\frac{3}{4}\sqrt{\frac{2}{3}}\,d_1 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 2\nu_1 \end{pmatrix}\begin{pmatrix} d_1 \\ -\frac{3}{4}\sqrt{\frac{2}{3}}\,d_1 \end{pmatrix}$$

Sustituyendo el valor del primer multiplicador $\nu_1 = -\sqrt{2/3}$ tenemos

$$\mathbf{w}^T \nabla^2_{\mathbf{xx}} \mathcal{L}\left(\mathbf{x}^*, \boldsymbol{\nu}^*\right) \mathbf{w} = \begin{pmatrix} d_1 & -\frac{3}{4}\sqrt{\frac{2}{3}}\,d_1 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & -2\sqrt{2/3} \end{pmatrix}\begin{pmatrix} d_1 \\ -\frac{3}{4}\sqrt{\frac{2}{3}}\,d_1 \end{pmatrix} =$$

$$= \begin{pmatrix} -\frac{3}{4}\sqrt{\frac{2}{3}}\,d_1 & 2d_1 \end{pmatrix}\begin{pmatrix} d_1 \\ -\frac{3}{4}\sqrt{\frac{2}{3}}\,d_1 \end{pmatrix} = -\frac{3}{4}\sqrt{\frac{2}{3}}\,d_1^2 - \frac{3}{2}\sqrt{\frac{2}{3}}\,d_1^2 =$$

$$= -\frac{9}{4}\sqrt{\frac{2}{3}}\,d_1^2 < 0 \quad \forall\, d_1 \neq 0$$

y la forma cuadrática $\mathbf{w}^T \nabla^2_{\mathbf{xx}} \mathcal{L}\left(\mathbf{x}^*, \boldsymbol{\nu}^*\right) \mathbf{w}$ es estrictamente cóncava. Dado que hablamos de un problema de maximización, podemos afirmar que se cumplen las condiciones suficientes de segundo orden para $\mathbf{x}^*$.

2

Optimización con restricciones: el caso infinito dimensional

Tras haber introducido y ejemplificado, en el primer capítulo, los resultados fundamentales de la teoría de la optimización con restricciones en el caso finito dimensional, pasaremos, en el tercer capítulo, a describir la teoría en el caso infinito dimensional. Esto nos llevará al calculo variacional y al cálculo de las ecuaciones de Euler-Lagrange asociadas a la minimización de un funcional de energía. Se trata de materia correspondiente, en el currículo de matemáticas, a un curso avanzado de análisis funcional. Antes, en este capítulo y en la sección 2.1, vamos a contextualizar el problema a resolver introduciendo conceptos y formulaciones propias de las redes neuronales, el aprendizaje profundo y la teoría del control. En la sección 2.2 formalizamos matemáticamente una clase de redes neuronales suficientemente grande para permitir la descripción de varias arquitecturas de redes. Entre ellas las ResNets. Veremos como estas redes se formulan como un paso de discretización de Euler explícito para la integración de las dinámicas continuas definidas por un sistema paramétrico de ecuaciones diferenciales ordinarias. Se trata de un paso conceptual muy importante, ya que define el concepto de *redes continuas* que evolucionan en forma de sistemas dinámicos, siendo cada arquitectura de red el resultado de una discretización adecuada del sistema.

En la sección 2.3 definimos el problema de optimización del aprendizaje profundo, lo que permite *entrenar* a la red a través del cálculo de los parámetros óptimos del problema de minimización. En la sección 2.4 interpretamos y formulamos el problema de aprendizaje profundo de una red en términos de la teoría del control. Para ello introducimos una clase general de controles admisibles formulando matemáticamente el sistema dinámico, cuya resolución mediante pasos de discretización de Euler explícito propaga hacia adelante y a través de las capas la información de los

datos. Finalizamos formalizando el problema de minimización del control óptimo, necesario para la estimación de los parámetros de la red.

2.1. Redes neuronales, aprendizaje profundo y teoría del control

El éxito reciente del desarrollo de las técnicas de aprendizaje profundo en aprendizaje automático (*machine learning*) y su aplicación en multitud de campos, disciplinas y problemas, ha generado un interés creciente por la comprensión de los fundamentos matemáticos que justifican estos resultados.

Una línea de investigación reciente y prometedora consiste en relacionar el aprendizaje profundo con la teoría del control óptimo, lo que permite definir la noción de un problema de aprendizaje continuo subyacente al marco discreto de las redes neuronales ($\mathcal{NN}$). En esta visión, las $\mathcal{NN}$ se pueden interpretar como una discretización de una EDO paramétrica que en el límite en el paso de discretización define una red neuronal continua en la profundidad, es decir, con un número infinito de capas. En este marco, las ResNets (*Deep residual learning for image recognition* [2]) se pueden considerar como la discretización progresiva de Euler (*forward Euler discretization*) de la formulación continua.

Una aplicación fundamental de este estudio consiste en proporcionar una manera eficiente para el entrenamiento de la red, ya que el entrenamiento de redes neuronales profundas es todavía una tarea desafiante.

El método más utilizado para el entrenamiento de las redes neuronales es el del descenso del gradiente estocástico SGD (Bottou, 2010 [11]) y sus variantes (Kingma y Ba, 2014 [12]), en donde las actualizaciones incrementales de los parámetros durante las épocas de entrenamiento se calculan utilizando la información proporcionada por el gradiente de la función de pérdida durante la fase de propagación retrógrada (*back-propagation*).

Este método tiene varios inconvenientes, ya que suele ser lento en las primeras etapas del descenso y puede quedarse atrapado en puntos de silla en donde el gradiente es prácticamente nulo, lo que se conoce como *vanishing gradient*. Si la tasa de aprendizaje no es lo suficientemente pequeña, tambien hay una patología debida a los gradientes elevados que explotan, es decir, tienden al infinito, lo que se conoce como *blowup* del sistema, con la consecuente pérdida de estabilidad y no convergencia.

Una alternativa para el entrenamiento supervisado consiste en formular el problema de aprendizaje como un problema de control óptimo y diseñar algoritmos basados en el principio del máximo de Pontryagin [13], véase Li y col. [14].

Este principio proporciona condiciones necesarias para la optimalidad de los parámetros de la red y se basa en dos componentes o ideas fundamentales: la definición de dinámicas hamiltonianas y la condición de que en cada instante de tiempo los parámetros óptimos maximizan el hamiltoniano.

Resumiendo, a partir de ahora vamos a dar un paso más. Al considerar un problema de optimización con restricciones, estas pasan a ser EDOs que tienen que cumplirse en un intervalo de tiempo finito, y en lugar de minimizar una función, lo que minimizaremos será un funcional de energía, es decir, una integral (energía), que, en última instancia, será la función de pérdida de la red.

En hipótesis de diferenciabilidad, la función de pérdida de la red permite el entrenamiento de la misma obteniendo los valores óptimos de los parámetros de la red que definen la respuesta (*output*) de la red en la fase de inferencia.

En este caso, que es la base de las arquitecturas de las ODENet, la restricción es satisfacer un **sistema dinámico** definido por una EDO vectorial, por lo que los multiplicadores de Lagrange ya no son vectores, sino que pasan a ser funciones que se llaman **coestados del sistema** o **adjuntos**. Utilizando el método del adjunto, para determinarlos se resuelve otra EDO, denominada *adjunta*. Son funciones medibles que cumplen con el **principio del máximo de Pontryagin**.

2.2. Arquitecturas de redes neuronales y aprendizaje profundo

En esta sección seguiremos el planteamiento y notación de Aghili y Mula en *Depth-adaptive neural networks from the optimal control viewpoint* [15]. Si bien es cierto que existen muchas arquitecturas de redes neuronales, vamos a simplificar la introducción y decir que las redes neuronales son clases de funciones o campos vectoriales de la forma general básica

$$\mathbf{v} : \mathbb{R}^n \to \mathbb{R}^k, \qquad \mathbf{x} \to \mathbf{v}(\mathbf{x}) = W_L\sigma(W_{L-1}\sigma(...\sigma(W_1(\mathbf{x})))$$

siendo L la profundidad, es decir, el número de capas de la red y para $l = 1...L$, las $W_l : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{k_l}$ son aplicaciones afines, siendo $n_0 = n$ y $n_L = k$.

Para todo $\mathbf{x} \in \mathbb{R}^{n_{l-1}}$ definimos $W_l(\mathbf{x}) = A_l(\mathbf{x}) + \mathbf{b}_l$ para una matriz $A_l \in \mathbb{R}^{n_l \times n_{l-1}}$ y $\mathbf{b}_l \in \mathbb{R}^{k_l}$, es decir

$$\mathbf{v}(\mathbf{x}) = W_L\sigma(W_{L-1}\sigma(...\sigma(W_1(\mathbf{x}))) = W_L\sigma(W_{L-1}\sigma(...\sigma(A_1(\mathbf{x}) + \mathbf{b}_1))$$

La función $\sigma : \mathbb{R} \to \mathbb{R}$ es una función de activación no lineal que actúa por componentes. Ejemplos típicos son la función Relu, la tangente hiperbólica $\sigma(x) = \tanh(x)$ y la función $\sigma(x) = \text{máx}\{0, x\}$ que se suele sustituir por la función *softmax*, que es una aproximante diferenciable (regularización) de la función *max*.

Con esta información definimos la clase de redes neuronales

$$\mathcal{NN}(L, \sigma, n_1, ...n_{l-1}) = \{\mathbf{v} : X \to Y / \, \mathbf{v}(\mathbf{x}) = W_L\sigma(W_{L-1}\sigma(...\sigma(W_1(\mathbf{x})))\}$$

Las funciones (redes) $\mathbf{v}(\mathbf{x})$ se pueden construir mediante composición repetida de funciones del tipo $\boldsymbol{\phi}_l : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$, $l = 1..L-1$ definidas por

$$\mathbf{x} \to \boldsymbol{\phi}_l(\mathbf{x}) = \sigma(W_l(\mathbf{x})) = \sigma(A_l(\mathbf{x}) + \mathbf{b}_l) \tag{2.1}$$

finalizando con una operación afín W_L y sin activación:

$$\mathbf{v} = W_L \circ \boldsymbol{\phi}_{L-1} \circ ... \circ \boldsymbol{\phi}_1, \qquad \forall \mathbf{v} \in \mathcal{NN}(L, \sigma, n_1, ...n_{l-1})$$

Esta simple formulación permite el diseño de varias arquitecturas de redes neuronales. El ejemplo más importante son las ResNets [2], que se obtienen definiendo $\boldsymbol{\phi}_l : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$, $l = 1..L-1$ mediante

$$\mathbf{x} \to \boldsymbol{\phi}_l(\mathbf{x}) = \mathbf{x} + h\sigma(W_l(\mathbf{x})) = \mathbf{x} + h\sigma(A_l(\mathbf{x}) + \mathbf{b}_l) \tag{2.2}$$

para un cierto parámetro $h > 0$ que identificaremos con un paso de discretización. Si fijamos $n_{L-1} = ... = n_1 = n_0 = n$, la repetida aplicación de $\boldsymbol{\phi}_l$ se puede ver como un paso de Euler explícito de $t_{l-1} = (l-1)h$ a $t_l = t_{l-1} + h$ para la integración de las dinámicas del sistema

$$\mathbf{x}'(t) = \mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}(t)) \tag{2.3}$$

siendo

$$\mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = \sigma(W_t(\mathbf{x}(t))) \qquad W_t(\mathbf{x})(t) = A_t\mathbf{x}(t) + \mathbf{b}_t$$

es decir

$$\mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = \sigma(A_t\mathbf{x}(t) + \mathbf{b}_t)$$

y donde los parámetros de la red son

$$\boldsymbol{\theta}(t) = \{A_t, \mathbf{b}_t\}$$

Podemos así ver una ResNet como la realización de $L-1$ pasos de discretización de Euler explícito de paso h de las dinámicas continuas definidas por (2.3) seguidos por un paso final $W_L : \mathbb{R}^k \to \mathbb{R}^k$. Utilizando distintos métodos numéricos de integración de (2.3) se obtienen otras arquitecturas de redes, como Polynet [16] o FractalNet [17], que se corresponden con la aplicación de un método de Euler implícito y otro de Runge-Kutta de segundo orden (RK2).

Señalamos finalmente una restricción del marco introducido, que radica en que las dimensiones del *input* y de cada una de las capas profundas tienen que ser iguales, $n_{L-1} = ... = n_1 = n_0 = n$. Por lo tanto, este marco no es aplicable a todas las redes neuronales, pero sí es adecuado para series temporales de datos.

2.3. Aprendizaje profundo

El aprendizaje profundo consiste en el entrenamiento de una red neuronal para obtener los parámetros óptimos de la misma que permitan hacer *inferencia* de los datos, es decir predicción, en forma de clasificación o regresión. Se trata de un problema de aprendizaje estadístico (*statistical learning problem*) que podemos formular como un problema de optimización con restricciones en el marco de la teoría del control óptimo.

2.3.1. El problema del aprendizaje estadístico

Sea $X \subset \mathbb{R}^n$ el conjunto de datos de entrenamiento y sea $Y \subset \mathbb{R}^k$ el conjunto de etiquetas (*label set*), siendo n, k enteros positivos. Supondremos la existencia de una distribución de probabilidad μ en el conjunto $X \times Y$ que representa la distribución de los datos, pares $(\mathbf{x}, \mathbf{y}) \in X \times Y$.

Dada una función de pérdida, $\mathcal{L} : Y \times Y \to \mathbb{R}^+$, el objetivo del problema de aprendizaje estadístico es determinar una función $\mathbf{v} : X \to Y$, llamada *prediction rule*, dentro de un conjunto (*hipothesis class*) $\mathcal{V} = \{\mathbf{v} : X \to Y\}$ tal que minimiza en $\mathcal{V}$ la pérdida esperada, o valor esperado de la pérdida, dada por

$$\mathcal{J}(\mathbf{v}) \doteq \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mu}\mathcal{L}(\mathbf{v}(\mathbf{x}), \mathbf{y})$$

El valor esperado de la función de pérdida $\mathcal{L}$ en un dataset con N muestras (ejemplos) se calcula como

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mu}\mathcal{L}(\mathbf{v}(\mathbf{x}), \mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(\mathbf{v}(x_i), y_i)$$

siendo $\mathbf{x}$ el vector de muestras, $\mathbf{v}(\mathbf{x})$ la predicción (*output*) de la red e $\mathbf{y}$ el vector de etiquetas. En otras palabras, tenemos el siguiente problema de optimización continuo:

Determinar una función $\mathbf{v}^* \in \mathcal{V}$ tal que

$$\mathbf{v}^* \in \arg\min_{\mathbf{v}\in\mathcal{V}} \mathcal{J}(\mathbf{v}) \tag{2.4}$$

El símbolo de pertenencia a un conjunto $\in$ se ha introducido al no haber, en general, una solución única al problema, ya que la topología de las redes es, típicamente, no convexa.

En la práctica no se conoce la distribución μ y tenemos un conjunto de N muestras $\mathbf{S}_N\{(x_i, y_i)\}_{i=0}^N$. La elección más típica es considerar una distribución uniforme[1] $\mu_N : X \times Y \to \mathbb{R}^+$:

$$\mu_N = \frac{1}{N}\sum_{i=1}^{N}\delta_{(x_i,y_i)}(\mathbf{x}, \mathbf{y})$$

que nos da la así llamada función de pérdida empírica (*empirical loss*)

$$\mathcal{J}_N(\mathbf{v}) \doteq \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mu}\mathcal{L}(\mathbf{v}(\mathbf{x}), \mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(\mathbf{v}(x_i), y_i)$$

a partir de la cual se calcula una aproximación $\mathbf{v}_N \in \mathcal{V}$ de $\mathbf{v}^*$ en (2.4) resolviendo el problema de optimización del aprendizaje profundo:

$$\mathbf{v}_N = \operatorname{argmín}_{\mathbf{v}\in\mathcal{V}} \mathcal{J}_N(\mathbf{v}) \tag{2.5}$$

[1] Todos los posibles valores que pueden adoptar las variables ($\mathbf{x}$, $\mathbf{y}$) tienen la misma probabilidad.

2.4. Teoría del control

Definida la estructura de las redes que se van a considerar, introducimos a continuación la notación necesaria para formular el problema de aprendizaje profundo de una red en términos de la teoría del control. Observamos que los parámetros de una red neuronal (pesos y sesgos) actúan como un control en la red. De hecho, determinan cómo el *input* se transforma a lo largo de las capas de la red para producir un *output* o predicción.

Definimos con $\boldsymbol{\Theta} \subset \mathbb{R}^m$ al conjunto de **controles admisibles** o pesos del entrenamiento (*learning weight*). Fijado $T > 0$ sea $L^\infty([0,T], \boldsymbol{\Theta})$ el conjunto de controles medibles esencialmente acotados que toman valores en $\boldsymbol{\Theta}$. Por ejemplo, diremos que $\boldsymbol{\theta} = \{\theta(t) : 0 \leq t \leq T\}$ si $\boldsymbol{\theta} \in L^\infty([0,T], \boldsymbol{\Theta})$.

Consideramos ahora las funciones f, campo vectorial que define las dinámicas hacia adelante (*feed-forward*) de la red; $\boldsymbol{\Phi}$, campo (escalar o vectorial) que define la función de pérdida final (*terminal loss*), y R, un campo escalar que definimos como regularizante sobre el conjunto de los controles admisibles:

$$f : \mathbb{R}^n \times \boldsymbol{\Theta} \to \mathbb{R}^n, \qquad \boldsymbol{\Phi} : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^k, \qquad R : \boldsymbol{\Theta} \to \mathbb{R}$$

Para cada control $\boldsymbol{\theta} \in L^\infty([0,T], \boldsymbol{\Theta})$ y cada valor de la variable aleatoria $\mathbf{x} \in X$ definimos las dinámicas de estado

$$\mathbf{u}^{\boldsymbol{\theta},\mathbf{x}} = \{u^{\boldsymbol{\theta},\mathbf{x}}(t) : 0 \leq t \leq T\}$$

como la solución de la EDO vectorial

$$\begin{cases} \dot{\mathbf{u}}^{\boldsymbol{\theta},\mathbf{x}}(t) = \quad f(\mathbf{u}^{\boldsymbol{\theta},\mathbf{x}}(t), \boldsymbol{\theta}(t)), \quad \forall\, t \in (0,T) \\ \mathbf{u}^{\boldsymbol{\theta},\mathbf{x}}(0) = \mathbf{x} \end{cases} \tag{2.6}$$

La EDO es estocástica, pero la única fuente de aleatoriedad está en la condición inicial. Con esta notación, el problema de aprendizaje profundo se puede formular como el problema de control óptimo de determinar

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta} \in L^\infty([0,T], \boldsymbol{\Theta})}{\arg\min} \mathcal{J}(\boldsymbol{\theta}) \qquad s.a. \quad (2.6) \tag{2.7}$$

siendo

$$\mathcal{J}(\boldsymbol{\theta}) \doteq \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mu}[\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})]$$

y en donde para cada par de datos de entrenamiento $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^k$ la función de pérdida es

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) = \underbrace{\boldsymbol{\Phi}(\mathbf{u}^{\boldsymbol{\theta},\mathbf{x}}(T), \mathbf{y})}_{Terminal\ cost} + \underbrace{\int_0^T R(\boldsymbol{\theta}) dt}_{Running\ cost}$$

siendo $(\boldsymbol{\Phi}(\mathbf{u}^{\boldsymbol{\theta},\mathbf{x}}(T), \mathbf{y})$ el coste debido a la desviación final del *output* de la etiqueta y $R(\boldsymbol{\theta})$ el coste de toda la trayectoria.

En la práctica, al no conocer la distribución μ, tenemos un conjunto de N muestras $\mathbf{S}_N\{(x_i, y_i)\}_{i=1}^N$. Dado un control $\boldsymbol{\theta} \in L^\infty([0,T], \boldsymbol{\Theta})$, cada muestra (x_i, y_i) sigue las dinámicas definidas por (2.6); lo que se traduce en

$$\begin{cases} \dot{\mathbf{u}}^{\boldsymbol{\theta},i}(t) = & f(\mathbf{u}^{\boldsymbol{\theta},i}(t), \boldsymbol{\theta}(t)), \quad \forall\, t \in (0,T) \\ \mathbf{u}^{\boldsymbol{\theta},i}(0) = \mathbf{x}_i & \qquad\qquad i = 1...N \end{cases} \tag{2.8}$$

2.4.1. El problema del control óptimo

En este problema se pide determinar

$$\boldsymbol{\theta}_{\mathbf{S}_N} = \arg\min_{\boldsymbol{\theta} \in L^\infty([0,T], \boldsymbol{\Theta})} \mathcal{J}_N(\boldsymbol{\theta}) \qquad s.a. \quad (2.8) \tag{2.9}$$

siendo la función de pérdida

$$\mathcal{J}_N(\boldsymbol{\theta}) \doteq \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mu}[\mathrm{Loss}(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})] = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\Phi}(\mathbf{u}^{\boldsymbol{\theta},i}(T), y_i) + \int_0^T R(\boldsymbol{\theta}) dt \tag{2.10}$$

La función $\boldsymbol{\Phi}$ representa la función de pérdida asociada a la salida de la red durante el entrenamiento (*terminal cost*) y tiene el papel de la función $\mathcal{L}$ introducida en la sección 2.3.1. El término de regularización R, *running cost*, podría depender también del estado del sistema $\mathbf{u}^{\boldsymbol{\theta},i}(t)$ en la forma $R(\mathbf{u}^{\boldsymbol{\theta},i}(t), \boldsymbol{\theta})$.

Observamos que las soluciones $\boldsymbol{\theta}_{\mathbf{S}_N}$ del problema de minimización de la función de pérdida empírica dependen del conjunto de muestra $\mathbf{S}_N$, por lo cual son variables aleatorias. Sin embargo, fijado $\mathbf{S}_N$, el problema es determinista y podemos utilizar la teoría del control óptimo.

La idea de base es que, como en todos los problemas de optimización considerados en esta memoria, existen condiciones necesarias para la existencia de soluciones del problema de control óptimo. En este caso se trata del principio del máximo de Pontryagin, que introduciremos más adelante. Previamente necesitamos introducir el **método del adjunto**. Este método nos permitirá calcular las variables adjuntas del sistema que tienen el papel de los multiplicadores de Lagrange asociados a la restricción de que el estado final del sistema tiene que minimizar el coste terminal.

3

El método del adjunto

Tras haber visto la posibilidad de formular una ResNet en términos de una red neuronal continua disctretizada, en este capítulo introduciremos los fundamentos matemáticos de la teoría y aplicación del **método del adjunto** para el cálculo del gradiente de un funcional a minimizar sujeto a restricciones dinámicas paramétricas definidas en las formulaciones del problema de control óptimo (2.9) y (2.10).

En primer lugar, en la sección 3.1, formalizaremos el problema de minimización con restricciones dinámicas que escribiremos en la sección 3.2, en forma de un problema sin restricciones a través de la definición de la función Lagrangiana aumentada, que incluye las dinámicas del sistema en téminos de una penalización.

A continuación, en la sección 3.11, expondremos los fundamentos matemáticos del método del adjunto que proponen Chen *et al* [1] en el contexto del entrenamiento supervisado de las redes neuronales y en sustitución parcial del paso de descenso del gradiente estocástico en la fase retrógada del cálculo del gradiente para minimización de la función de pérdida.

Finalmente, sentadas las bases teóricas y formalismos matemáticas necesarios, nos apoyaremos en problemas modelo que ilustrarán esta teoría y su aplicación. Concretamente, veremos dos ejemplos de aplicación de la teoría para facilitar su comprensión. Para ello, nos hemos apoyado en el artículo *PDE-constrained optimization and the adjoint method*, de Andrew M. Bradley [18].

En el capítulo 4, basádonos en el libro de Evans, *An Introduction to Mathematical Optimal Control Theory* [19], introduciremos y aplicaremos, a través del método del adjunto, el principio del máximo de Pontryagin. Las condiciones necesarias definidas en el teorema de Pontryagin permiten calcular un estado y un coestado del sistema con los cuales obtener un control óptimo (conjunto de parámetros óptimos de la red) del sistema dinámico discretizado que modela la red neuronal.

3.1. El problema de minimización con restricciones dinámicas

Sea f un campo escalar, $f : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$ que queremos minimizar y $\mathbf{g}$ un campo vectorial $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}^n$ que define un sistema de ecuaciones (restricciones).

Siguiendo el texto y la notación de Bradley en *Pde-constrained optimization and the adjoint method* [18], empezamos estableciendo el marco de trabajo definido por el problema general de optimización paramétrica con restricciones

$$\min_{\mathbf{p}} f(\mathbf{x}, \mathbf{p}), \qquad \text{sujeto a} \quad \mathbf{g}(\mathbf{x}, \mathbf{p}) = \mathbf{0} \tag{3.1}$$

donde $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{p} \in \mathbb{R}^d$ y n y d son números enteros positivos. Comparar con (2.8), (2.9) y (2.10).

Podemos ver que la diferencia entre (3.1) y el problema de optimización con restricciones definido en (1.1) es la aparición de los parámetros $\mathbf{p}$ en (3.1). Además, la minimización se realiza sobre el conjunto de parámetros admisibles. Esto permitirá establecer una analogía con la teoría del control óptimo para sistemas dinámicos, en donde identificaremos los parámetros óptimos (de la red neuronal) con controles óptimos del sistema.

La variable $\mathbf{x}$ contiene los valores de la variable de campo (presión, temperatura, velocidad, etc) que describen el **estado del sistema**. La variable $\mathbf{p}$ denota un conjunto de parámetros que podrían parametrizar condiciones de contorno e iniciales para una EDP discretizada o condiciones iniciales para una EDO.

Las restricciones $\mathbf{g}(\mathbf{x}, \mathbf{p}) = \mathbf{0}$ pueden surgir al semidiscretizar (discretizar en el espacio y no en el tiempo) una EDP o, como hemos visto, al modelar una red neuronal residual (ResNet) mediante EDO.

Este modelado confiere la posibilidad de definir una red continua y, al mismo tiempo, generaliza el marco anterior al considerar las soluciones del sistema, que son funciones en lugar de puntos en el espacio $\mathbb{R}^n$. El espacio de las soluciones es *infinito dimensional* y es necesario utilizar técnicas del cálculo variacional para encontrarlas.

El problema de minimización en $\mathbf{p}$ para $f(\mathbf{x}, \mathbf{p})$ se suele llamar el **problema inverso** y la resolución en $\mathbf{x}$ de la ecuación $\mathbf{g}(\mathbf{x}, \mathbf{p}) = \mathbf{0}$, el **problema directo**[1]. Típicamente, el problema (3.1) es un problema de optimización *finito dimensional* que requiere una resolución numérica debido a las no linealidades que aparecen en f y $\mathbf{g}$. Sin embargo, para los casos de minimización cuadrática y restricciones de EDO lineales veremos que la potencia de cálculo simbólico de programas como Matlab, Octave o Python permite la resolución analítica.

La representación gráfica servirá para entender el proceso de minimización con restricciones.

[1] *inverse problem* y *forward problem* en la literatura anglosajona.

Finalmente, consideraremos problemas que pueden depender del tiempo en las restricciones, lo que permite una mayor generalidad al incluir restricciones que son EDO no autónomas.

3.2. El lagrangiano aumentado

Tal y como vimos en la sección 1.4, es posible reformular el problema (3.1) como un problema de minimización sin restricciones considerando la función lagrangiana aumentada

$$\mathcal{L}(\mathbf{x}, \mathbf{p}) = f(\mathbf{x}, \mathbf{p}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}, \mathbf{p})$$

siendo $\boldsymbol{\lambda}$ el vector de **multiplicadores de Lagrange**. En el marco del problema (3.1), consideramos el problema particular de minimización paramétrica con restricciones dinámicas

$$\min_{\mathbf{p}} F(\mathbf{x}, \mathbf{p}) \tag{3.2}$$

sujeto a $\quad \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{p}, t) = \mathbf{0}, \qquad \mathbf{g}(\mathbf{x}(0), \mathbf{p}) = \mathbf{0},$

siendo

$$F(\mathbf{x}, \mathbf{p}) = \int_0^T f(\mathbf{x}, \mathbf{p}, t) dt, \tag{3.3}$$

un funcional[2] de tipo integral y en donde

$$\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{p}, t) = 0 \tag{3.4}$$

es un sistema de EDOs en forma implícita y

$$\mathbf{g}(\mathbf{x}(0), \mathbf{p}) = \mathbf{0}$$

es un vector de condiciones iniciales que modelan las restricciones del problema de minimización y que es función de los parámetros desconocidos $\mathbf{p}$.

Suponiendo suficiente regularidad (la aplicabilidad del teorema de la función implícita a la función $\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{p}, t)$), el sistema de EDO en forma explícita se obtiene explicitando las derivadas $\dot{\mathbf{x}}$ en la función $\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{p}, t)$, en la forma

$$\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{p}, t) = \dot{\mathbf{x}} - \mathbf{h}(\mathbf{x}, \mathbf{p}, t)$$

de donde se obtiene el sistema explícito

$$\dot{\mathbf{x}} = \mathbf{h}(\mathbf{x}, \mathbf{p}, t) \tag{3.5}$$

Las EDO consideradas hasta el momento, en las redes neuronales son del tipo (3.5). El marco teórico propuesto permite, sin embargo, considerar también EDOs del tipo implícito (3.4).

[2]Entendemos por funcional un campo escalar definido por un operador, en este caso el operador integral, que se aplica sobre funciones y no sobre puntos del espacio. De manera no formal, un funcional es una función de funciones.

La solución del sistema de EDO es una función vectorial $\mathbf{x}(t) \doteq \mathbf{x}(t, \mathbf{p}) \in \mathbb{R}^n$ que representa el **estado del sistema** en el tiempo t. Depende además de una función vectorial $\mathbf{p}(t) \in \mathbb{R}^d$ cuyos valores óptimos en cada instante se obtienen resolviendo el problema de minimización paramétrica con restricciones dinámicas (3.2). Más adelante identificaremos la función vectorial $\mathbf{p}(t)$ como los controles óptimos del sistema dinámico definido por (3.5). En este sentido, el problema (3.2) es un caso especial de una clase general de problemas de control óptimo para EDO.

La consecuencia es que podemos formular el problema del aprendizaje supervisado de las redes neuronales de tipo ResNet en los términos más generales de un problema de control óptimo en espacios funcionales (infinito dimensionales). Para ello, necesitaremos utilizar el cálculo variacional.

En este contexto, un algoritmo de minimización basado en el gradiente necesita el cálculo de todas las derivadas parciales en términos de los parámetros para la determinación de sus valores óptimos.

Suponiendo la regularidad suficiente para llevar a cabo las operaciones indicadas, aplicamos la regla de la cadena para obtener que, en términos de los parámetros:

$$d_{\mathbf{p}} F(\mathbf{x}, \mathbf{p}) = \int_0^T [\partial_{\mathbf{x}} f \, d_{\mathbf{p}} \mathbf{x} + \partial_{\mathbf{p}} f] dt \tag{3.6}$$

El cálculo de $d_{\mathbf{p}}\mathbf{x}$, es decir, de las variaciones de la solución de la EDO en términos de los parámetros es, en muchos casos, complicado. Sin embargo, es posible evitarlo mediante el método del adjunto.

3.3. El método del adjunto

El método del adjunto consiste en definir un **coestado del sistema**, o **estado adjunto**, como la solución de una EDO retrógrada en el tiempo que tiene una forma adjunta a la EDO progresiva que define las dinámicas (estados) del sistema. Para ello, se deduce una EDO para el cálculo del vector adjunto $\boldsymbol{\lambda}$, lo que permitirá obtener el gradiente $d_{\mathbf{p}} F(\mathbf{x}, \mathbf{p})$. El trabajo total de calcular F y su gradiente es equivalente a resolver dos sistemas de EDO, uno para $\mathbf{x}(t)$ durante la fase de propagación hacia adelante (*feed-forward*) y otro, el adjunto, para el cálculo del coestado $\boldsymbol{\lambda}(t)$ durante la fase de retro-propagación (*backward propagation*).

El primer paso para la resolución de problemas del tipo (3.2) consiste en escribir la función lagrangiana aumentada para el problema de minimización sin restricciones

$$\mathcal{L}(\mathbf{x}, \mathbf{p}) \doteq \int_0^T [f(\mathbf{x}, \mathbf{p}, t) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{p}, t)] dt + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}(0), \mathbf{p}) \tag{3.7}$$

donde el vector de multiplicadores de Lagrange $\boldsymbol{\lambda}$ es una función del tiempo, $\boldsymbol{\lambda} = \boldsymbol{\lambda}(t)$ y define el coestado del sistema. El vector de multiplicadores $\boldsymbol{\mu}$ se ha introducido en el funcional para dar cuenta de las condiciones iniciales del sistema de EDO. Se observa que en el conjunto de funciones admisibles (las que satisfacen las restricciones

en el problema (3.2)) los funcionales F en (3.3) y el lagrangiano $\mathcal{L}$ en (3.7) coinciden y, por tanto, coinciden sus gradientes:

$$d_{\mathbf{p}}F(\mathbf{x},\mathbf{p}) = d_{\mathbf{p}}\mathcal{L}(\mathbf{x},\mathbf{p})$$

Utilizando la regla de la cadena calculamos el gradiente de $\mathcal{L}$ para obtener:

$$\begin{aligned} d_{\mathbf{p}}\mathcal{L}(\mathbf{x},\mathbf{p}) = \int_0^T \left[\partial_{\mathbf{x}} f\, d_{\mathbf{p}}\mathbf{x} + \partial_{\mathbf{p}} f + \boldsymbol{\lambda}^T \left(\partial_{\mathbf{x}}\mathbf{h}\, d_{\mathbf{p}}\mathbf{x} + \partial_{\dot{\mathbf{x}}}\mathbf{h}\, d_{\mathbf{p}}\dot{\mathbf{x}} + \partial_{\mathbf{p}}\mathbf{h}\right)\right] dt + \\ + \boldsymbol{\mu}^T \left(\partial_{\mathbf{x}(0)}\mathbf{g}\, d_{\mathbf{p}}\mathbf{x}(0) + \partial_{\mathbf{p}}\mathbf{g}\right). \end{aligned} \tag{3.8}$$

El integrando contiene términos en $d_{\mathbf{p}}\mathbf{x}$ y en $d_{\mathbf{p}}\dot{\mathbf{x}}$. Integrando por partes el término $\boldsymbol{\lambda}^T \partial_{\dot{\mathbf{x}}}\mathbf{h}\, d_{\mathbf{p}}\dot{\mathbf{x}}$ en (3.8) y escribiendo la diferencial en la forma (usamos la notación $d\mathbf{x}(t) = \dot{\mathbf{x}}(\mathbf{t})dt$)

$$(d_{\mathbf{p}}\dot{\mathbf{x}})\, dt = d(d_{\mathbf{p}}\mathbf{x})$$

obtenemos

$$\int_0^T \boldsymbol{\lambda}^T\, \partial_{\dot{\mathbf{x}}}\mathbf{h}\, d_{\mathbf{p}}\dot{\mathbf{x}}\, dt = \boldsymbol{\lambda}^T\, \partial_{\dot{\mathbf{x}}}\mathbf{h}\, d_{\mathbf{p}}\mathbf{x}\big|_0^T - \int_0^T \left[\dot{\boldsymbol{\lambda}}^T \partial_{\dot{\mathbf{x}}}\mathbf{h} + \boldsymbol{\lambda}^T \frac{d}{dt}\left(\partial_{\dot{\mathbf{x}}}\mathbf{h}\right)\right] d_{\mathbf{p}}\mathbf{x}\, dt$$

Sustituyendo este resultado en (3.8) y agrupando términos en $d_{\mathbf{p}}\mathbf{x}$ y en $d_{\mathbf{p}}\mathbf{x}(0)$ tenemos:

$$\begin{aligned} d_{\mathbf{p}}\mathcal{L}(\mathbf{x},\mathbf{p}) = \underbrace{\int_0^T \left[\left(\partial_{\mathbf{x}} f + \boldsymbol{\lambda}^T \left(\partial_{\mathbf{x}}\mathbf{h} - \frac{d}{dt}\left(\partial_{\dot{\mathbf{x}}}\mathbf{h}\right)\right) - \dot{\boldsymbol{\lambda}}^T \partial_{\dot{\mathbf{x}}}\mathbf{h}\right) d_{\mathbf{p}}\mathbf{x} + \partial_{\mathbf{p}} f + \boldsymbol{\lambda}^T \partial_{\mathbf{p}}\mathbf{h}\right] dt}_{T_1} + \\ + \underbrace{\boldsymbol{\lambda}^T \partial_{\dot{\mathbf{x}}}\mathbf{h}\, d_{\mathbf{p}}\mathbf{x}\big|_{t=T}}_{T_2} + \underbrace{\left(-\boldsymbol{\lambda}^T \partial_{\dot{\mathbf{x}}}\mathbf{h} + \boldsymbol{\mu}^T \mathbf{g}_{\mathbf{x}(0)}\right)\big|_{t=0} d_{\mathbf{p}}\mathbf{x}(0)}_{T_3} + \underbrace{\boldsymbol{\mu}^T \partial_{\mathbf{p}}\mathbf{g}}_{T_4}. \end{aligned}$$

Como ya hemos mencionado, el cálculo del término diferencial $d_{\mathbf{p}}\mathbf{x}(T)$ que aparece en T_2 es complejo. Por tanto, fijamos la condición de tiempo final

$$\boldsymbol{\lambda}(T) = \mathbf{0} \tag{3.9}$$

para cancelar todo el término T_2. Para cancelar el penúltimo término (T_3) podemos fijar

$$\boldsymbol{\mu}^T = \boldsymbol{\lambda}^T \partial_{\dot{\mathbf{x}}}\mathbf{h}\big|_{t=0}\, \mathbf{g}_{\mathbf{x}(0)}^{-1} \tag{3.10}$$

Por último, también podemos evitar calcular $d_{\mathbf{p}}\mathbf{x}$ en el término T_1 en tiempos $t > 0$ fijando

$$\partial_{\mathbf{x}} f + \boldsymbol{\lambda}^T \left(\partial_{\mathbf{x}}\mathbf{h} - \frac{d}{dt}\left(\partial_{\dot{\mathbf{x}}}\mathbf{h}\right)\right) - \dot{\boldsymbol{\lambda}}^T \partial_{\dot{\mathbf{x}}}\mathbf{h} = 0$$

que reordenamos para obtener la EDO adjunta en la forma

$$\dot{\boldsymbol{\lambda}}^T \partial_{\dot{\mathbf{x}}}\mathbf{h} = \boldsymbol{\lambda}^T \left(\partial_{\mathbf{x}}\mathbf{h} - \frac{d}{dt}\left(\partial_{\dot{\mathbf{x}}}\mathbf{h}\right)\right) + \partial_{\mathbf{x}} f \tag{3.11}$$

siendo $\boldsymbol{\lambda}(t)$ el coestado o estado adjunto del sistema. También introducimos las definiciones y conceptos de estado de un sistema y problemas directos e inversos.

Finalizamos esta sección con el pseudocódigo del algoritmo para calcular el gradiente $d_{\mathbf{p}}F(\mathbf{x},\mathbf{p})$.

Pseudocódigo del algoritmo del cálculo del gradiente (método del adjunto)

1. Integramos $\mathbf{h}(\mathbf{x},\dot{\mathbf{x}},\mathbf{p},t)=\mathbf{0}$ en $\mathbf{x}$ de $t=0$ a T con condición inicial vectorial $\mathbf{g}(\mathbf{x}(0),\mathbf{p})=\mathbf{0}$.
2. Integramos la EDO adjunta (3.11)

$$\partial_{\mathbf{x}}f+\boldsymbol{\lambda}^T\left(\partial_{\mathbf{x}}\mathbf{h}-\frac{d}{dt}\left(\partial_{\dot{\mathbf{x}}}\mathbf{h}\right)\right)-\dot{\boldsymbol{\lambda}}^T\partial_{\dot{\mathbf{x}}}\mathbf{h}=0$$

   en $\boldsymbol{\lambda}(t)$ de $t=T$ a 0 (alternativa a *backpropagation*) con condición final

$$\boldsymbol{\lambda}(T)=0$$

   Para transformar el problema de valores finales (PVF) adjunto en un problema de valores iniciales (PVI) se define un tiempo $\tau=T-t$, con $d\tau=-dt$ y se resuelve el problema para $\boldsymbol{\lambda}(\tau)$.
3. Calculamos

$$d_{\mathbf{p}}F(\mathbf{x},\mathbf{p})=\int_0^T\left[\partial_{\mathbf{p}}f+\boldsymbol{\lambda}^T\partial_{\mathbf{p}}\mathbf{h}\right]dt+\boldsymbol{\lambda}^T\partial_{\dot{\mathbf{x}}}\mathbf{h}\big|_{t=0}\,\mathbf{g}_{\mathbf{x}(0)}^{-1}\,\partial_{\mathbf{p}}\mathbf{g}$$

Podemos ver que al integrar en el paso 1 obtenemos $\mathbf{x}$, y cuando integramos en el paso 2 obtenemos $\boldsymbol{\lambda}$. Así podemos calcular la derivada de F respecto a los parámetros.

3.4. Ejemplos

Se ha incluido en el anexo B.0.3 un ejemplo simple que puede ser resuelto de forma analítica y que permite entender la aplicación del método del adjunto.

Ahora vamos a plantear un ejemplo más complejo de aplicación del método del adjunto al calculo del gradiente de un funcional a minimizar sujeto a restricciones dinámicas paramétricas dadas por un sistema de EDOs.

Ejemplo 3.4.1 Regresión lineal sujeta a restricciones dinámicas. Sea $F(\mathbf{x},\mathbf{p})$ una función paramétrica dada, cuyo comportamiento queremos estudiar cuando sus variables verifican ciertas restricciones dinámicas. En concreto, sea

$$F(\mathbf{x},\mathbf{p})=\left(\frac{1}{2}\sum_{r=1}^{N}\int_0^T|x_r(t)-d_r(t)|^2dt\right),\quad\text{sujeta a}\quad\begin{cases}\dot{\mathbf{x}}=b\mathbf{x}\\ \mathbf{x}(0)-\mathbf{a}=\mathbf{0}\end{cases}$$

con

$$\mathbf{x} = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \qquad \mathbf{d} = \begin{pmatrix} \sin(t) \\ \cos(t) \end{pmatrix}$$

siendo $\mathbf{x}$ el estado del sistema y $\mathbf{d}$ unos datos que definen una serie temporal de datos (tras discretización del sistema). Se pide calcular el gradiente $d_{\mathbf{p}}F(\mathbf{x}, \mathbf{p})$ siendo $\mathbf{p} = [\mathbf{a}, b]^T \in \mathbb{R}^3$ y estudiar la sensibilidad del problema frente a variaciones en los parámetros del sistema $\mathbf{p}$.

Observamos que la función depende implícitamente de los parámetros, ya que aparecen en el sistema pero no en la expresión de $F(\mathbf{x}, \mathbf{p})$. Fijado un punto $\mathbf{p} \in \mathbb{R}^3$, existe una única trayectoria en el plano de fases $\mathbf{x} = \mathbf{x}(\mathbf{p})$ con energía de mínimos cuadrados $F(\mathbf{x}, \mathbf{p}) = F(\mathbf{x}(\mathbf{p}), \mathbf{p}) = F(\mathbf{p}) \in \mathbb{R}$. Notamos además que $N = 2$, ya que el vector $\mathbf{x}$ tiene dos componentes.

En este caso las dimensiones del problema son $N = 2$, $d = 3$, ya que tenemos 2 EDO y 3 parámetros ($\mathbf{p} = [a_1, a_2, b] = [\mathbf{a}, b]^T \in \mathbb{R}^3$) en el sistema dinámico. Por la linealidad del operador integral podemos intercambiar el símbolo de sumatorio con el símbolo de integración y definir la función (o lagrangiano)

$$f(\mathbf{x}, \mathbf{p}, t) = \frac{1}{2}\sum_{r=1}^{N} |x_r(t) - d_r(t)|^2 = \frac{1}{2}\|\mathbf{x}(t) - \mathbf{d}(t)\|^2$$

siendo el sistema dinámico definido por las EDO y condiciones iniciales

$$\mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{p}, t) = \dot{\mathbf{x}} - b\mathbf{x}, \qquad \mathbf{g}(\mathbf{x}(0), \mathbf{p}) = \mathbf{x}(0) - \mathbf{a}$$

Definimos mediante (3.7) el lagrangiano aumentado para el problema de minimización sin restricciones para la variable $\mathbf{x}(t)$, el estado adjunto $\boldsymbol{\lambda}(t)$ y los multiplicadores de Lagrange $\boldsymbol{\mu}$.

$$\mathcal{L}(\mathbf{x}, \mathbf{p}) \doteq \int_0^T [f(\mathbf{x}, \mathbf{p}, t) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{p}, t)]dt + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}(0), \mathbf{p}) =$$

$$= \int_0^T \left(\frac{1}{2}\|\mathbf{x}(t) - \mathbf{d}(t)\|^2 + \boldsymbol{\lambda}^T(\dot{\mathbf{x}} - b\mathbf{x})\right) dt + \boldsymbol{\mu}^T(\mathbf{x}(0) - \mathbf{a})$$

Siguiendo los pasos indicados en 3.3:

1. Resolución del PVI. Como en el ejemplo anterior, las EDO que definen este problema son EDO lineales de primer orden y separables, por lo que su resolución es directa. Al integrar las EDO obtenemos

$$\mathbf{x}(t) = \begin{pmatrix} a_1 e^{bt} \\ a_2 e^{bt} \end{pmatrix}$$

Para $b < 0$, el origen en el plano de fases $(x_1(t), x_2(t))$ es un punto de equilibrio asintótico (para tiempos grandes, es decir, en el límite $t \to \infty$). Sin embargo el horizonte temporal de este problema es $T > 0$ finito.

2. A partir de las ecuaciones

$$f(\mathbf{x},\mathbf{p},t)=\frac{1}{2}\sum_{r=1}^{N}|x_r(t)-d_r(t)|^2,\qquad h(\mathbf{x},\dot{\mathbf{x}},\mathbf{p},t)=\dot{\mathbf{x}}-b\mathbf{x}$$

calculamos las derivadas parciales

$$\partial_{\mathbf{x}}f=\left(a_1e^{b\,t}-\sin(t),a_2e^{b\,t}-\cos(t)\right),$$

$$\partial_{\mathbf{x}}\mathbf{h}=\begin{pmatrix}-b & 0\\ 0 & -b\end{pmatrix},\qquad \partial_{\dot{\mathbf{x}}}\mathbf{h}=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

que nos permiten calcular la EDO adjunta

$$\partial_{\mathbf{x}}f+\boldsymbol{\lambda}^T\left(\partial_{\mathbf{x}}\mathbf{h}-\frac{d}{dt}\left(\partial_{\dot{\mathbf{x}}}\mathbf{h}\right)\right)-\dot{\boldsymbol{\lambda}}\,\partial_x\mathbf{h}=\mathbf{0}$$

siendo $\boldsymbol{\lambda}^T=(\lambda_1(t),\lambda_2(t))^T$ el estado adjunto del sistema. La EDO se complementa con las condiciones finales $\boldsymbol{\lambda}(T)=\mathbf{0}$. Sustituyendo las expresiones calculadas se tiene el problema de valores finales:

$$(PVF)\begin{cases}\begin{pmatrix}a_1e^{bt}-\sin(t)\\ a_2e^{bt}-\cos(t)\end{pmatrix}^T-\boldsymbol{\lambda}^T\begin{pmatrix}-b & 0\\ 0 & -b\end{pmatrix}-\dot{\boldsymbol{\lambda}}^T\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}=\mathbf{0};\\ \\ \text{sujeto a }\boldsymbol{\lambda}(T)=\mathbf{0},\end{cases}$$

que se trata de un sistema de ecuaciones adjuntas de primer orden lineales y de coeficientes constantes. Puede ser resuelto de manera analítica o simbólica. Tras calcularlo simbólicamente con el código de Python, tenemos

$$\boldsymbol{\lambda}(t)=\begin{pmatrix}-e^{-bt}\left[\dfrac{a_1\left(b^2e^{2bT}-b^2e^{2bt}+e^{2bT}-e^{2bt}\right)}{2b\left(b^2+1\right)}+\right.\\ \left.+\dfrac{2b\left(-be^{bT}\sin(T)+be^{bt}\sin(t)+e^{bT}\cos(T)-e^{bt}\cos(t)\right)}{2b\left(b^2+1\right)}\right]\\ \\ -e^{-bt}\left[\dfrac{a_2\left(b^2e^{2bT}-b^2e^{2bt}+e^{2bT}-e^{2bt}\right)}{2b\left(b^2+1\right)}+\right.\\ \left.+\dfrac{2b\left(-be^{bT}\cos(T)+be^{bt}\cos(t)-e^{bT}\sin(T)+e^{bt}\sin(t)\right)}{2b\left(b^2+1\right)}\right]\end{pmatrix}$$

3. Calculamos $d_{\mathbf{p}}F(\mathbf{x},\mathbf{p})$. Tenemos que

$$\partial_{\mathbf{p}}f=(0,0,0),\qquad \partial_{\mathbf{p}}\mathbf{h}=\begin{pmatrix}0 & 0 & -x_1(t)\\ 0 & 0 & -x_2(t)\end{pmatrix}$$

$$\mathbf{g}_{\mathbf{x}(0)}=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix},\qquad \partial_{\mathbf{p}}\mathbf{g}=\begin{pmatrix}-1 & 0 & 0\\ 0 & -1 & 0\end{pmatrix}$$

Por tanto, obtenemos el gradiente de $F(\mathbf{x}, \mathbf{p})$ como sigue

$$d_{\mathbf{p}}F(\mathbf{x}, \mathbf{p}) = \int_0^T \left[\partial_{\mathbf{p}} f + \boldsymbol{\lambda}^T \partial_{\mathbf{p}} \mathbf{h}\right] dt + \boldsymbol{\lambda}^T \partial_{\dot{\mathbf{x}}} \mathbf{h}\big|_{t=0} \, \mathbf{g}_{\mathbf{x}(0)}^{-1} \, \partial_{\mathbf{p}} \mathbf{g}$$

Como $\partial_{\mathbf{p}} f = (0, 0, 0)$,

$$d_{\mathbf{p}}F(\mathbf{x}, \mathbf{p}) = \int_0^T \boldsymbol{\lambda}^T(t) \begin{pmatrix} 0 & 0 & -x_1(t) \\ 0 & 0 & -x_2(t) \end{pmatrix} dt +$$

$$+\boldsymbol{\lambda}^T(0) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} =$$

$$= \int_0^T \boldsymbol{\lambda}^T(t) \begin{pmatrix} 0 & 0 & -x_1(t) \\ 0 & 0 & -x_2(t) \end{pmatrix} dt + \boldsymbol{\lambda}^T(0) \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} =$$

$$= \int_0^T (0, 0, -x_1(t)\lambda_1(t) - x_2(t)\lambda_2(t)) \; dt - (\lambda_1(0), \lambda_2(0), 0)$$

Denotamos $d_{\mathbf{p}}F = (d_{a_1}F, d_{a_2}F, d_bF)$. Al conocer el estado del sistema y su adjunto se sustituye para obtener las expresiones explícitas de las derivadas parciales en función de los parámetros del sistema

$$d_{a_1}F = \frac{a_1\left(b^2e^{2bT} - b^2 + e^{2bT} - 1\right) - 2b\left(be^{bT}\operatorname{sen}(T) - e^{bT}\cos(T) + 1\right)}{2b\left(b^2+1\right)}$$

$$d_{a_2}F = \frac{a_2\left(b^2e^{2bT} - b^2 + e^{2bT} - 1\right) - 2b\left(be^{bT}\cos(T) - b + e^{bT}\operatorname{sen}(T)\right)}{2b\left(b^2+1\right)}$$

$$d_bF = \frac{v}{4b^2\left(b^2+1\right)^2}$$

siendo $v = a_1\left(a_1b^4 + 2a_1b^2 + a_1 + 8b^3\right) - a_1e^{bT}\left(-2Ta_1b^5e^{bT} - 4Ta_1b^3e^{bT} + \right.$

$-2Ta_1be^{bT} + 4Tb^5\operatorname{sen}(T) - 4Tb^4\cos(T) + 4Tb^3\operatorname{sen}(T) - 4Tb^2\cos(T) +$

$\left. + a_1b^4e^{bT} + 2a_1b^2e^{bT} + a_1e^{bT} - 4b^4\operatorname{sen}(T) + 8b^3\cos(T) + 4b^2\operatorname{sen}(T)\right) +$

$+a_2\left(a_2b^4 + 2a_2b^2 + a_2 - 4b^4 + 4b^2\right) - a_2e^{bT}\left(-2Ta_2b^5e^{bT} - 4Ta_2b^3e^{bT} + \right.$

$-2Ta_2be^{bT} + 4Tb^5\cos(T) + 4Tb^4\operatorname{sen}(T) + 4Tb^3\cos(T) + 4Tb^2\operatorname{sen}(T) +$

$\left. + a_2b^4e^{bT} + 2a_2b^2e^{bT} + a_2e^{bT} - 4b^4\cos(T) - 8b^3\operatorname{sen}(T) + 4b^2\cos(T)\right)$

Las expresiones anteriores son algo complicadas para su análisis. Sin embargo la representación gráfica permite entender cualitativamente el comportamiento del sistema.

Los siguientes gráficos ilustrativos se han obtenido mediante el código de python que puede encontrarse en el anexo.

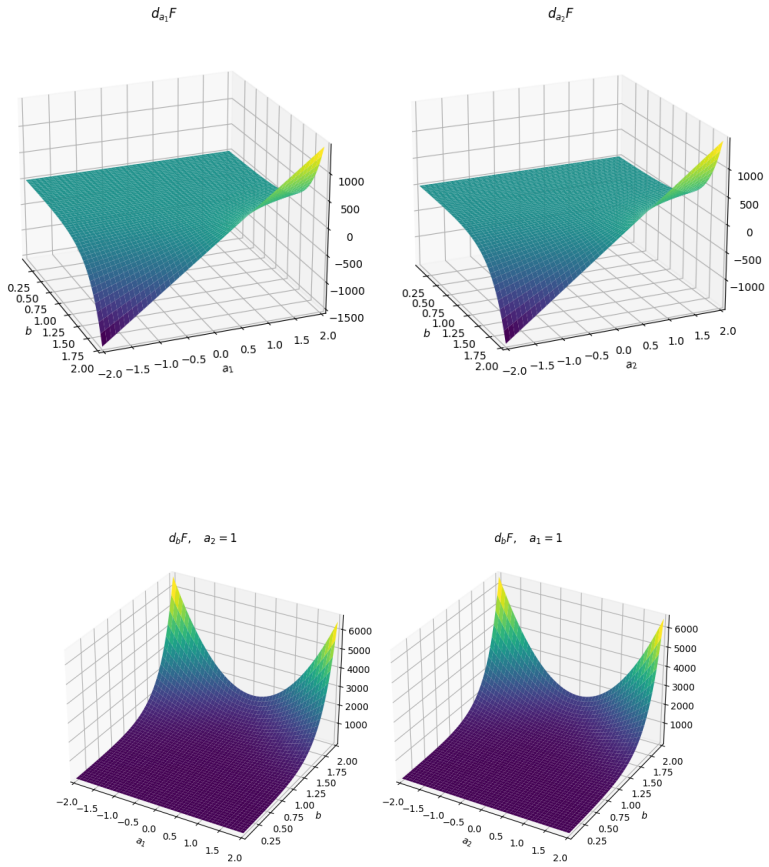

Figura 3.1: Gráficos del cálculo del gradiente del ejemplo 3.4.1 y análisis de sensibilidad del problema con respecto a los parámetros del sistema. Se ha representado cada componente del gradiente respecto a sus variables, siempre considerando $T = 2$. En la primera fila se observa que para valores de b pequeños, los parámetros (a_1, a_2) (es decir las condiciones iniciales) apenas tienen efecto. Para valores mayores de b el sistema responde a las variaciones. En la segunda fila podemos ver que para b cercanas a 0, el gradiente se mantiene estable, pero en cuanto aumentamos el valor de b, el valor del gradiente aumenta considerablemente.

4

Principio del máximo de Pontryagin

En este capítulo nos vamos a servir del capítulo 4 de *An Introduction to Mathematical Optimal Control Theory* de Evans [19] para abordar las ideas generales del principio del máximo de Pontryagin, que nos va a servir de broche final para cerrar todos los contenidos teóricos estudiados. Para ampliar los conocimientos relacionados con este capítulo se puede consultar el libro de Evans.

El principio del máximo de Pontryagin generaliza las condiciones necesarias de KKT para optimización con restricciones en optimización infinito dimensional y permite la determinación de la trayectoria óptima de un sistema dinámico, es decir, que evoluciona de forma continua con el tiempo. Veremos que, si existe un control óptimo, entonces existe una función llamada coestado del sistema o adjunto que, junto con el estado del sistema y un control óptimo, verifica este principio. La técnica se basa en el cálculo del hamiltoniano de un sistema dinámico (3.5) que permite escribir las ecuaciones de Euler-Lagrange del problema de optimización de primer orden para las variables vectoriales $\mathbf{x}(t) =$ estado, $\mathbf{p}(t) =$ adjunto y $\boldsymbol{\alpha}(t) =$ control del sistema.

En la sección 4.4 formularemos el problema básico del cálculo variacional y relacionaremos las ecuaciones de Euler-Lagrange del funcional de energía a minimizar con la teoría de los sistemas dinámicos hamiltonianos. La condición necesaria de optimalidad dada por las ecuaciones de Euler-Lagrange se define en el teorema 4.1.2. Las dinámicas hamiltonianas se describen en el teorema 4.1.5.

En la sección 4.2 introduciremos, enunciaremos y ejemplificaremos el principio del máximo de Pontryagin, que se aplica en el contexto de los sistemas dinámicos y la teoría del control. Se trata de una generalización de la teoría de la optimalidad de KKT. Si existe un control óptimo para el sistema, entonces existe un coestado del sistema verificando una EDO vectorial adjunta que maximiza el hamiltoniano del sistema. El cálculo del coestado a través de la resolución de la EDO adjunta realiza

la fase de retro-propagación (*backprop*) de la red. No hay necesidad de almacenar en memoria los estados y parámetros de la red a la salida de cada capa, sino que es suficiente resolver la EDO adjunta en tiempo retrógrado.

Finalizaremos el capítulo y la memoria en la sección 4.3, con un ejemplo modelo que permite entender los pasos fundamentales de la teoría expuesta.

4.1. Cálculo de variaciones, dinámica hamiltoniana

La relación entre las ecuaciones de Euler-Lagrange y los sistemas hamiltonianos, una conexión fundamental en mecánica clásica y cálculo de variaciones, radica en que ambas formulaciones describen las dinámicas de sistemas físicos y están relacionadas a través de la transformada de Legendre.

Vamos a comenzar esta sección con una introducción a algunos métodos variacionales que nos van a servir de motivación para el principio del máximo de Pontryagin.

Las ecuaciones de Euler-Lagrange se derivan a partir del principio de la acción estacionaria. Este principio afirma que la trayectoria, en el espacio de fases, solución del sistema es tal que la acción (integral del lagrangiano en el tiempo) es estacionaria, es decir, tiene un mínimo con respecto a las variaciones de la trayectoria.

Podemos formular el problema básico del cálculo de variaciones.

Definición 4.1.1 Sea $L : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, $L = L(\mathbf{x}, \mathbf{v})$ una función suave[1] siendo L el lagrangiano o función lagrangiana del sistema. Sean $T > 0$, x^0, $x^1 \in \mathbb{R}^n$ dados. El problema básico del cálculo de variaciones consiste en: determinar una curva $\mathbf{x}^*(\cdot) : [0, T] \to \mathbb{R}^n$ que minimice el funcional

$$I\left[\mathbf{x}^*(\cdot)\right] \doteq \min_{\mathbf{x}} \int_0^T L\left(\mathbf{x}(t), \dot{\mathbf{x}}(t)\right) dt \tag{4.1}$$

de entre todas las funciones $\mathbf{x}^*(\cdot)$ que satisfacen $\mathbf{x}(0) = \mathbf{x}^0$ y $\mathbf{x}(T) = \mathbf{x}^1$. Nótese que, a partir de (4.1) se tiene la caracterización de la curva (trayectoria) óptima

$$\mathbf{x}^* = \operatorname*{argmín}_{\mathbf{x}} \int_0^T L\left(\mathbf{x}(t), \dot{\mathbf{x}}(t)\right) dt \tag{4.2}$$

Ahora asumamos que $\mathbf{x}^*(\cdot)$ resuelve nuestro problema variacional. La pregunta fundamental es: ¿cómo podemos caracterizar $\mathbf{x}^*(\cdot)$?

[1] Entendemos por *suave* a una función de clase C^1.

4.1.1. Las ecuaciones de Euler-Lagrange

Tal y como vimos en los problemas de optimización finitos dimensionales, la caracterización del óptimo del problema es a través de las condiciones necesarias de I orden que se obtienen derivando el funcional en términos de la curva.

NOTACIÓN. Para ello escribimos $L = L(\mathbf{x}, \mathbf{v})$, considerando la variable $\mathbf{x}$ como la posición y la variable $\mathbf{v}$ como la velocidad. Las derivadas parciales de L son:

$$\frac{\partial L}{\partial x_i} = L_{x_i}, \qquad \frac{\partial L}{\partial v_i} = L_{v_i} \quad (1 \leq i \leq n)$$

y escribimos los respectivos gradientes

$$\nabla_{\mathbf{x}} L \coloneqq (L_{x_1}, \ldots, L_{x_n}), \quad \nabla_{\mathbf{v}} L \coloneqq (L_{v_1}, \ldots, L_{v_n})$$

Teorema 4.1.2 ***Ecuaciones de Euler-Lagrange (E-L)***
Sea $\mathbf{x}^*(\cdot)$*, definida en* (4.2)*, solución del problema de cálculo de variaciones. Entonces* $\mathbf{x}^*(\cdot)$ *resuelve las ecuaciones diferenciales de Euler-Lagrange:*

$$\frac{d}{dt}\left[\nabla_{\mathbf{v}} L\left(\mathbf{x}^*(t), \dot{\mathbf{x}}^*(t)\right)\right] = \nabla_{\mathbf{x}} L\left(\mathbf{x}^*(t), \dot{\mathbf{x}}^*(t)\right) \tag{E-L}$$

La importancia de este teorema es que si podemos resolver las ecuaciones de Euler-Lagrange, entonces la solución de nuestro problema original de cálculo de variaciones, si existe, estará entre las soluciones de dichas ecuaciones. La condición es, por tanto, **necesaria**. En hipótesis de convexidad del funcional (al ser un problema de miniimización) la condición es **suficiente**. Es importante observar que (E-L) es un sistema de n EDO de segundo orden. La ecuación i-ésima de ese sistema es:

$$\frac{d}{dt}\left[L_{v_i}\left(\mathbf{x}^*(t), \dot{\mathbf{x}}^*(t)\right)\right] = L_{x_i}\left(\mathbf{x}^*(t), \dot{\mathbf{x}}^*(t)\right)$$

La demostración de este teorema se encuentra en la sección 4.1 de *An Introduction to Mathematical Optimal Control Theory* de Evans [19].

4.1.2. Conversión a ecuaciones hamiltonianas

En mecánica clásica las variables que definen la configuración de un sistema se llaman **coordenadas generalizadas**, y el espacio definido por esas coordenadas se llama el **espacio de configuraciones** (estados) del sistema físico. Típicamente se denota por $\mathbf{q} = \mathbf{q}(t)$ a un punto en el espacio de configuraciones. Esta convención se mantiene en la formulación hamiltoniana de la mecánica clásica, en donde se usa la variable $\mathbf{p}$ para denotar al **momento generalizado** (o **momento conjugado**) para sustituir a la variable de velocidad generalizada $\dot{\mathbf{q}} = d\mathbf{q}/dt$, $\dot{\mathbf{q}} = \dot{\mathbf{q}}(t)$ típica de

la mecánica lagrangiana. Utilizando esta notación, se tendría

$$L\left(\mathbf{x}(t), \dot{\mathbf{x}}(t)\right) = L\left(\mathbf{q}(t), \dot{\mathbf{q}}(t)\right)$$

donde no hay dependencia explícita del tiempo, indicando así que el sistema es autónomo.

Para facilitar el seguimiento del texto de Evans [19], seguiremos utilizando las variables $\mathbf{x}$, $\dot{\mathbf{x}}$ en lugar de las coordenadas generalizadas $\mathbf{q}$, $\dot{\mathbf{q}}$ más propias de la física-matemática.

Definición 4.1.3 Para la curva $\mathbf{x}(\cdot)$, llamamos **momento conjugado** a

$$\mathbf{p}(t) \coloneqq \nabla_{\mathbf{v}} L\left(\mathbf{x}(t), \dot{\mathbf{x}}(t)\right) \qquad (0 \leq t \leq T)$$

Vamos a reescribir las ecuaciones de Euler-Lagrange, de II orden, como un sistema de EDO de primer orden para $\mathbf{x}(\cdot)$, $\mathbf{p}(\cdot)$. Para ello es necesario definir el **hamiltoniano** del sistema. Asumimos por hipótesis que, para todo $\mathbf{x}, \mathbf{p} \in \mathbb{R}^n$, podemos resolver la ecuación

$$\mathbf{p} = \nabla_{\mathbf{v}} L(\mathbf{x}, \mathbf{v}) \tag{4.3}$$

para $\mathbf{v}$ en función de $\mathbf{x}$ y $\mathbf{p}$. Es decir, suponemos que podemos resolver (4.3) para $\mathbf{v} = \mathbf{v}(\mathbf{x}, \mathbf{p})$.

Definición 4.1.4 Definimos el **hamiltoniano** $H : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ del sistema como el campo escalar dado por la fórmula

$$H(\mathbf{x}, \mathbf{p}) = \mathbf{p} \cdot \mathbf{v}(\mathbf{x}, \mathbf{p}) - L(\mathbf{x}, \mathbf{v}(\mathbf{x}, \mathbf{p})) \tag{4.4}$$

con $\mathbf{v}$ definida como en nuestra hipótesis.

La definición del hamiltoniano se obtiene aplicando la **transformada de Legendre** definida en (4.4) al lagrangiano. Digamos que la transformada de Legendre permite pasar del formalismo lagrangiano al formalismo hamiltoniano en mecánica clásica, es decir, newtoniana.

NOTACIÓN. Denotamos las derivadas parciales de H por

$$\frac{\partial H}{\partial x_i} = H_{x_i}, \qquad \frac{\partial H}{\partial p_i} = H_{p_i} \quad (1 \leq i \leq n)$$

y escribimos

$$\nabla_{\mathbf{x}} H \coloneqq \left(H_{x_1}, \ldots, H_{x_n}\right), \quad \nabla_{\mathbf{p}} H \coloneqq \left(H_{p_1}, \ldots, H_{p_n}\right)$$

Teorema 4.1.5 ***Dinámica hamiltoniana***
Sea $\mathbf{x}(\cdot)$ solución de las ecuaciones (E-L) y sea $\mathbf{p}(\cdot)$ el momento generalizado definido en 4.1.3. Entonces el par $(\mathbf{x}(\cdot), \mathbf{p}(\cdot))$ resuelve las ecuaciones de Hamilton:

$$\begin{cases} \dot{\mathbf{x}}(t) &= \nabla_p H\left(\mathbf{x}(t), \mathbf{p}(t)\right) \\ \dot{\mathbf{p}}(t) &= -\nabla_{\mathbf{x}} H\left(\mathbf{x}(t), \mathbf{p}(t)\right) \end{cases} \tag{H}$$

Además, el mapeo $t \mapsto H(\mathbf{x}(t), \mathbf{p}(t))$ *es constante a lo largo de las órbitas soluciones del sistema.*

De nuevo, la demostración de este teorema la encontramos en la página 44 del texto de Evans [19].

Recuperamos aqui la demostración de la conservatividad del sistema de Hamilton (H). En sistemas conservativos donde no existe disipación ni generación de energía, la energía total del sistema se conserva en el tiempo. Esto se expresa matemáticamente en la forma $dH/dt = 0$. Aplicando la regla de la cadena y suponiendo la regularidad suficiente se obtiene

$$\frac{d}{dt}H(\mathbf{x}(t), \mathbf{p}(t)) = \nabla_{\mathbf{x}}H \cdot \dot{\mathbf{x}}(t) + \nabla_{\mathbf{p}}H \cdot \dot{\mathbf{p}}(t) = \nabla_{\mathbf{x}}H \cdot \nabla_{\mathbf{p}}H + \nabla_{\mathbf{p}}H \cdot (-\nabla_{\mathbf{x}}H) = 0.$$

Las ecuaciones de Hamilton describen la evolución del sistema en términos de ecuaciones de I orden y el teorema afirma que las trayectorias soluciones del sistema dinámico son curvas de nivel del hamiltoniano H.

4.2. El principio del máximo de Pontryagin

Pasamos a la formulación del principio del máximo de Pontryagin.

Este teorema establece que si $\boldsymbol{\alpha}^*(\cdot)$ es un control óptimo, entonces existe una función $\mathbf{p}^*(\cdot)$ denominada *coestado* del sistema que satisface cierto principio de maximización. Debemos pensar en $\mathbf{p}^*(\cdot)$ como una especie de multiplicador de Lagrange que aparece debido a la restricción de que la curva óptima $\mathbf{x}^*(\cdot)$ debe satisfacer un sistema de EDO. Al igual que los multiplicadores de Lagrange convencionales son útiles para resolver problemas de optimización con restricciones, también lo será el coestado, siendo en este caso la restricción ser solución de un sistema dinámico.

Como dijo Francis Clarke: «El principio del máximo fue, de hecho, la culminación de una larga búsqueda en el cálculo de variaciones de una regla multiplicativa exhaustiva, que es la manera correcta de verlo: $p(t)$ es un "multiplicador de Lagrange" [...] Hace del control óptimo una herramienta de diseño, mientras que el cálculo de variaciones era una manera de estudiar la naturaleza.» [19].

Pasamos a formular el problema básico de la teoría del control óptimo.

Definición 4.2.1 Dado un funcional de recompensa $P : \mathcal{A} \to \mathbb{R}$, determinar un control $\boldsymbol{\alpha}^*(\cdot)$ tal que

$$P[\boldsymbol{\alpha}^*(\cdot)] = \max_{\boldsymbol{\alpha}(\cdot)\in\mathcal{A}} P[\boldsymbol{\alpha}(\cdot)] \tag{4.5}$$

siendo $\mathcal{A}$ el conjunto de funciones dado por los controles admisibles.

Esta teoría se puede aplicar suponiendo $\boldsymbol{\alpha}(\cdot) \in \mathcal{A}$ funciones medibles según Lebesgue. Lo típico es considerar controles esencialmente acotados, $\boldsymbol{\alpha}(\cdot) \in L^\infty(0, T)$.

4.2.1. Problema de tiempo libre, punto final fijo

Introducimos la formulación básica de un problema de control de tiempo libre pero con punto final fijo. Dado un control $\boldsymbol{\alpha}(\cdot) \in \mathcal{A}$, resolvemos para la correspondiente evolución de nuestro sistema:

$$\begin{cases} \dot{\mathbf{x}}(t) &= \mathbf{f}\left(\mathbf{x}(t), \boldsymbol{\alpha}(t)\right) \quad (t \geq 0) \\ \mathbf{x}(0) &= \mathbf{x}^0 \end{cases} \tag{EDO}$$

Asumamos que nos dan un punto objetivo $\mathbf{x}^1 \in \mathbb{R}^n$. Es decir, queremos llevar las dinámicas del sistema a un punto (estado) fijo independientemente del tiempo necesario para ello. Entonces introducimos el funcional de recompensa

$$P\left[\boldsymbol{\alpha}(\cdot)\right] = \int_0^\tau r\left(\mathbf{x}(t), \boldsymbol{\alpha}(t)\right) dt \tag{P}$$

El lagrangiano $r : \mathbb{R}^n \times \mathcal{A} \to \mathbb{R}$ es la recompensa corriente y $\tau = \tau\left[\boldsymbol{\alpha}(\cdot)\right] \leq \infty$ denota la primera vez que la solución de (EDO) llega al punto objetivo $\mathbf{x}^1$. Es un horizonte de tiempo libre.

El problema de optimización será encontrar un control óptimo $\boldsymbol{\alpha}^*(\cdot)$ tal que

$$P\left[\boldsymbol{\alpha}^*(\cdot)\right] = \max_{\boldsymbol{\alpha}(\cdot) \in \mathcal{A}} P\left[\boldsymbol{\alpha}(\cdot)\right] \doteq \max_{\boldsymbol{\alpha}(\cdot) \in \mathcal{A}} \int_0^\tau r\left(\mathbf{x}(t), \boldsymbol{\alpha}(t)\right) dt$$

El principio del máximo de Pontryagin (PMP), escrito abajo, afirma la existencia de una función $\mathbf{p}^*(\cdot)$ que, junto con la trayectoria óptima $\mathbf{x}^*(\cdot)$, satisface un análogo al sistema hamiltoniano (H). Para esto necesitamos un hamiltoniano adecuado para el marco de la teoría del control.

Definición 4.2.2 El **hamiltoniano de la teoría del control** es la función (campo escalar)

$$H(\mathbf{x}, \mathbf{p}, \boldsymbol{\alpha}) := \mathbf{f}(\mathbf{x}, \boldsymbol{\alpha}) \cdot \mathbf{p} + r(\mathbf{x}, \boldsymbol{\alpha}) \quad (\mathbf{x}, \mathbf{p} \in \mathbb{R}^n, \boldsymbol{\alpha} \in \mathcal{A})$$

Si comparamos con la definición clásica de hamiltoniano 4.1.4, ecuación (4.4) vemos que hay un cambio de signo debido a que el problema de optimización en teoría del control óptimo es de maximización.

Pasamos al enunciado del principio del máximo de Pontryagin.

Teorema 4.2.3 ***Principio del máximo de Pontryagin***
Sea $\boldsymbol{\alpha}^*(\cdot)$ óptimo para (EDO), (P) *y $\mathbf{x}^*(\cdot)$ la trayectoria correspondiente. Entonces existe una función $\mathbf{p}^* : [0, \tau^*] \to \mathbb{R}^n$ tal que*

$$\dot{\mathbf{x}}^*(t) = \nabla_p H\left(\mathbf{x}^*(t), \mathbf{p}^*(t), \boldsymbol{\alpha}^*(t)\right) \tag{ODE}$$

$$\dot{\mathbf{p}}^*(t) = -\nabla_x H\left(\mathbf{x}^*(t), \mathbf{p}^*(t), \boldsymbol{\alpha}^*(t)\right) \tag{ADJ}$$

y

$$H\left(\mathbf{x}^*(t), \mathbf{p}^*(t), \boldsymbol{\alpha}^*(t)\right) = \max_{\boldsymbol{\alpha} \in \mathcal{A}} H\left(\mathbf{x}^*(t), \mathbf{p}^*(t), \boldsymbol{\alpha}\right) \quad (0 \leq t \leq \tau^*) \tag{M}$$

Además,

$$H\left(\mathbf{x}^*(t), \mathbf{p}^*(t), \boldsymbol{\alpha}^*(t)\right) \equiv 0 \quad (0 \leq t \leq \tau^*)$$

Donde τ^* denota la primera vez que la trayectoria $\mathbf{x}^*(\cdot)$ alcanza el punto objetivo $\mathbf{x}^1$. Llamamos a $\mathbf{x}^*(\cdot)$ estado del sistema de control óptimo y a $\mathbf{p}^*(\cdot)$ el coestado.

OBSERVACIÓN Y ADVERTENCIA. Más precisamente, deberíamos definir

$$H(\mathbf{x}, \mathbf{p}, q, \boldsymbol{\alpha}) = \mathbf{f}(\mathbf{x}, \boldsymbol{\alpha}) \cdot \mathbf{p} + qr(\mathbf{x},\boldsymbol{\alpha}) \qquad q \in \mathbb{R}.$$

Una enunciación más cuidadosa del principio del máximo establece que «existe una constante $q \geq 0$ y una función $\mathbf{p}^* : [0, t^*] \to \mathbb{R}^n$ tal que (ODE), (ADJ) y (M) se mantienen». Si $q > 0$, podemos renormalizar para obtener $q = 1$ como hemos hecho antes. Si $q = 0$, entonces H no depende de la recompensa corriente r y en este caso el principio del máximo de Pontryagin no es útil. Esto es comúnmente denominado *problema anormal*.

La demostración de este teorema se encuentra en la sección A.5 del apéndice del texto de Evans [19].

4.3. Ejemplo: aterrizaje lunar

Con el ejemplo del aterrizaje lunar vamos a repasar todo lo expuesto en este último capítulo. Veremos que el cálculo del adjunto permite determinar la forma del control óptimo del sistema, es decir, los parámetros óptimos de la red. Así, podremos decir que se ha transformado una red discreta(ResNet) en un problema de control óptimo para EDO, es decir, la red continua ODENet.

El ejemplo puede encontrarse en *An Introduction to Mathematical Optimal Control Theory* de Evans [19], y para su resolución se han consultado tanto este libro como el artículo de Gazzola y Marchini: *The moon lander optimal control problem revisited* [20].

El objetivo es aterrizar una nave espacial en la superficie lunar con el menor gasto posible de combustible.

Introducimos la notación:

$$\begin{aligned} h(t) &= \text{altura en el tiempo } t \\ v(t) &= \text{velocidad } = \dot{h}(t) \\ m(t) &= \text{masa de la nave (cambia según se quema el combustible)} \\ \alpha(t) &= \text{empuje en el tiempo } t \end{aligned}$$

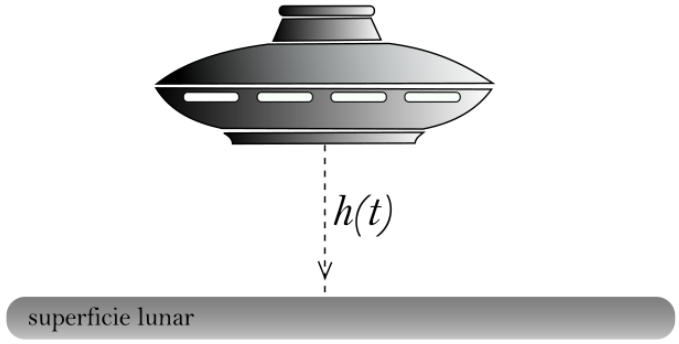

Figura 4.1: Ilustración del ejemplo: moon lander.

La función $\alpha(t)$ es el control, y asumimos que $0 \leq \alpha(T) \leq 1$; es decir, $A = [0, 1]$. Si $\alpha(t) = 0$, entonces el motor está apagado y la nave se encuentra en caída libre, mientras que si $\alpha(t) = 1$ el motor está encendido y se está empleando el máximo empuje contra la gravedad. Conforme se quema el combustible, la masa $m(t)$ de la nave disminuye con el tiempo y la tasa de cambio es negativamente proporcional a $\alpha(t)$. El balance de masa es $\dot{m}(t) = -k\alpha(t)$, que se complementa con la segunda ley de Newton para el equilibrio

$$m(t)\ddot{h}(t) = -gm(t) + \alpha(t)$$

siendo el lado derecho de la ecuación la diferencia entre la fuerza gravitacional y el empuje de la nave. Se trata de una EDO de II orden que se puede escribir como un sistema de EDO de primer orden.

El sistema modelo resultante es no lineal y acoplado; se define por la EDO vectorial

$$\begin{cases} \dot{h}(t) &= v(t) \\ \dot{v}(t) &= -g + \dfrac{\alpha(t)}{m(t)} \\ \dot{m}(t) &= -k\alpha(t) \end{cases} \qquad \text{(EDO)}$$

con condiciones iniciales

$$\begin{cases} h(0) &= h_0 > 0 \\ v(0) &= v_0 \\ m(0) &= m_0 > m_s \end{cases}$$

Podemos resumir estas ecuaciones de la forma

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \alpha(t))$$

para $\mathbf{x}(t) = (v(t), h(t), m(t))$.

Queremos asegurar un aterrizaje seguro minimizando la cantidad de combustible utilizado, es decir, maximizando el combustible restante al aterrizar. Por tanto, buscamos un control que maximice un funcional de recompensa (*payoff*) P

$$\max_{\alpha} P\left[\alpha(\cdot)\right] = m(\tau) \tag{P}$$

siendo $m(\tau)$ la masa residual, es decir, aquella que queda al aterizar de forma segura, y donde τ denota la primera vez que $h(\tau) = v(\tau) = 0$. Este es un problema de punto final variable, ya que el tiempo final no se conoce de antemano. También tenemos las restricciones para soluciones físicamente admisibles

$$h(t) \geq 0, \qquad m(T) \geq 0$$

ya que ni la altura del cohete ni su masa pueden ser menores que 0.

Dado que $\alpha(t) = -\dfrac{\dot{m}(t)}{k}$, nuestro objetivo es equivalente a minimizar el empuje total aplicado antes del aterrizaje, es decir

$$\int_0^\tau \alpha(t)dt = \frac{m_0 - m(\tau)}{k}$$

con lo que definimos

$$P\left[\alpha(\cdot)\right] = -\int_0^\tau \alpha(t)dt = \frac{m(\tau) - m_0}{k}$$

En términos de notación general tenemos

$$\mathbf{x}(t) = \begin{pmatrix} h(t) \\ v(t) \\ m(t) \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} v \\ -g + \alpha/m \\ -k\alpha \end{pmatrix}$$

Por tanto, el hamiltoniano es

$$\begin{aligned} H(x, \mathbf{p}, \alpha) &= \mathbf{f} \cdot \mathbf{p} + r \\ &= (v, -g + \alpha/m, -k\alpha) \cdot (p_1, p_2, p_3) - \alpha \\ &= -\alpha + p_1 v + p_2 \left(-g + \frac{\alpha}{m}\right) + p_3(-k\alpha) \end{aligned}$$

Ahora tenemos que obtener las dinámicas del adjunto. Para nuestro hamiltoniano

$$H_{x_1} = H_h = 0, \quad H_{x_2} = H_v = p_1, \quad H_{x_3} = H_m = -\frac{p_2\alpha}{m^2}$$

Por tanto

$$\left\{\begin{array}{rcl} \dot{p}^1(t) &=& 0 \\ \dot{p}^2(t) &=& -p^1(t) \\ \dot{p}^3(t) &=& \dfrac{p^2(t)\alpha(t)}{m(t)^2}. \end{array}\right. \tag{ADJ}$$

La condición de maximización es

$$\begin{aligned} H(\mathbf{x}(t),\mathbf{p}(t),\alpha(t)) &= \max_{0\le\alpha\le 1} H(\mathbf{x}(t),\mathbf{p}(t),\alpha) \\ &= \max_{0\le\alpha\le 1}\left\{-\alpha + p^1(t)v(t) + p^2(t)\left[-g+\frac{\alpha}{m(t)}\right] + p^3(t)(-k\alpha)\right\} \\ &= p^1(t)v(t) - p^2(t)g + \max_{0\le\alpha\le 1}\left\{\alpha\left(-1+\frac{p^2(t)}{m(t)} - kp^3(t)\right)\right\} \end{aligned} \tag{M}$$

Por tanto, la regla del control óptimo viene dada por la regla:

$$\alpha(t) = \left\{\begin{array}{ll} 1 & \text{si } -1+\frac{p^2(t)}{m(t)} - kp^3(t) > 0 \\ 0 & \text{si } -1+\frac{p^2(t)}{m(t)} - kp^3(t) < 0 \end{array}\right.$$

Usando el principio del máximo. Vamos a tratar de averiguar la forma de la solución, y a comprobar que cumple con el principio del máximo. Empezamos suponiendo que primero dejamos apagado el motor ($\alpha \equiv 0$) y encendemos el motor solo al final. Denotamos como τ la primera vez que $h(\tau) = v(\tau) = 0$, cuando la nave aterriza. Suponemos que existe un tiempo de cambio $t^* < \tau$ cuando encendemos los motores a máxima potencia (fijando $\alpha \equiv 1$). Por tanto,

$$\alpha(t) = \left\{\begin{array}{ll} 0 & \text{para } 0 \le t \le t^* \\ 1 & \text{para } t^* \le t \le \tau. \end{array}\right.$$

Así, para tiempos $t^* \le t \le \tau$ nuestra EDO se convierte en

$$\left\{\begin{array}{rcll} \dot{h}(t) &=& v(t) & \\ \dot{v}(t) &=& -g + \dfrac{1}{m(t)} & (t^* \le t \le \tau) \\ \dot{m}(t) &=& -k & \end{array}\right.$$

con $h(\tau) = 0, v(\tau) = 0, m(t^*) = m_0$. Resolvemos el sistema integrando en el intervalo (t, τ) para obtener:

$$\begin{cases} m(t) &= m_0 + k(t^* - t) \\ v(t) &= g(\tau - t) + \dfrac{1}{k} \log \left[\dfrac{m_0 + k(t^* - \tau)}{m_0 + k(t^* - t)} \right] \\ h(t) &= -\dfrac{g(t-\tau)^2}{2} - \dfrac{m_0}{k^2} \log \left[\dfrac{m_0 + k(t - \tau)}{m_0} \right] + \dfrac{t - \tau}{k} \end{cases}$$

Si igualamos $t = t^*$:

$$\begin{cases} m(t^*) &= m_0 \\ v(t^*) &= g(\tau - t^*) + \dfrac{1}{k} \log \left[\dfrac{m_0 + k(t^* - \tau)}{m_0} \right] \\ h(t^*) &= -\dfrac{g(t^*-\tau)^2}{2} - \dfrac{m_0}{k^2} \log \left[\dfrac{m_0 + k(t^* - \tau)}{m_0} \right] + \dfrac{t^* - \tau}{k} \end{cases}$$

Supongamos que la cantidad total de combustible al empezar es m_1; por lo que $m_0 - m_1$ es el peso de la nave vacía. Cuando $\alpha \equiv 1$, el combustible se consume a razón k. Por tanto

$$k(\tau - t^*) \leq m_1$$

y entonces $0 \leq \tau - t^* \leq \dfrac{m_1}{k}$. Antes de t^*, fijamos $\alpha \equiv 0$.

Por tanto, (EDO) se convierte en

$$\begin{cases} \dot{h} &= v \\ \dot{v} &= -g \\ \dot{m} &= 0 \end{cases}$$

y, por tanto

$$\begin{cases} m(t) &= m_0 \\ v(t) &= -gt + v_0 \\ h(t) &= -\dfrac{1}{2} g t^2 + t v_0 + h_0 \end{cases}$$

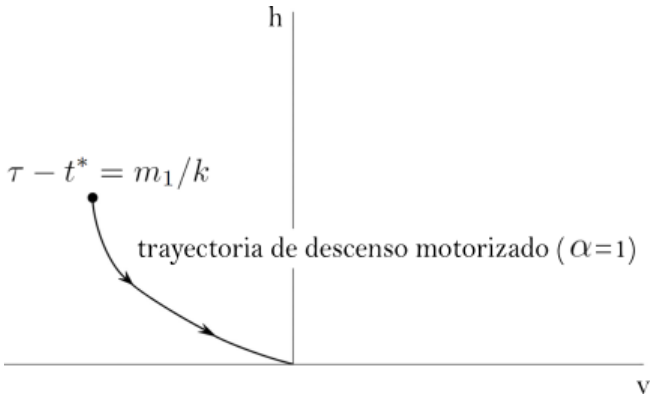

Figura 4.2: Trayectoria de la aeronave en descenso motorizado.

Combinamos las fórmulas de $v(t)$ y $h(t)$ para descubrir

$$h(t) = h_0 - \frac{1}{2g}\left(v^2(t) - v_0^2\right), \qquad 0 \le t \le t^*$$

Deducimos que la trayectoria en caída libre en el plano $(v(t), h(t))$ se encuentra en una parábola

$$h(v(t)) = h_0 - \frac{1}{2g}\left(v^2(t) - v_0^2\right), \qquad 0 \le t \le t^*$$

Si después nos movemos a lo largo de la parábola hasta que llegamos a la curva de aterrizaje suave de la figura anterior, podemos encender el motor del cohete y aterrizar de manera segura.

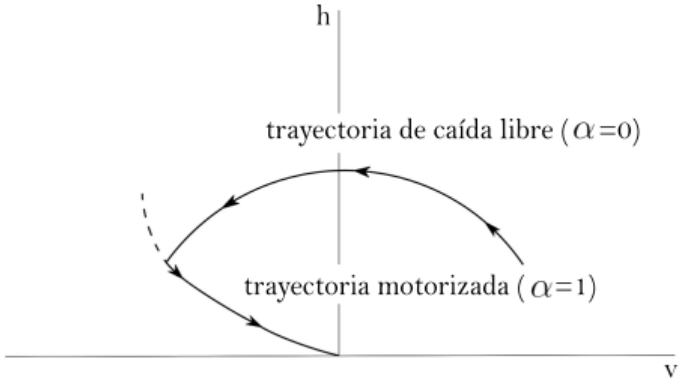

Figura 4.3: Trayectoria de la aeronave en caída libre.

En el siguiente caso ilustrado nos saltamos la curva de cambio y, por tanto, no podremos aterrizar de manera segura cambiando solo una vez.

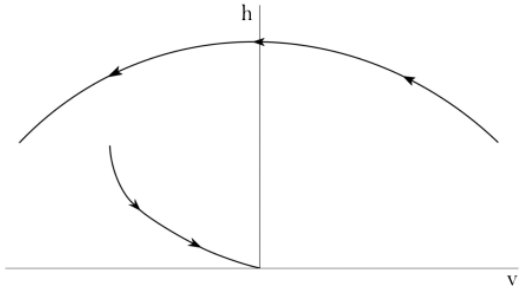

Figura 4.4: Trayectoria de la aeronave sin alcanzar la curva de cambio.

Para justificar nuestra deducción sobre la estructura del control óptimo, vamos a encontrar el coestado $\mathbf{p}(\cdot)$ tal que las funciones $\alpha(\cdot)$ y $\mathbf{x}(\cdot)$ descritas anteriormente satisfacen (EDO), (ADJ) y (M). Para hacer esto tendremos que averiguar las condiciones iniciales apropiadas

$$p^1(0) = \lambda_1,\ p^2(0) = \lambda_2,\ p^3(0) = \lambda_3$$

Resolvemos (ADJ) para $\alpha(\cdot)$ y tenemos

$$\begin{cases} p^1(t) \equiv & \lambda_1 & (0 \le t \le \tau) \\ p^2(t) = & \lambda_2 - \lambda_1 t & (0 \le t \le \tau) \\ p^3(t) = & \begin{cases} \lambda_3 & (0 \le t \le t^*) \\ \lambda_3 + \displaystyle\int_{t^*}^{t} \frac{\lambda_2 - \lambda_1 s}{\left(m_0 + k\left(t^* - s\right)\right)^2} ds & (t^* \le t \le \tau) \end{cases} \end{cases}$$

Definimos

$$r(t) \coloneqq 1 - \frac{p^2(t)}{m(t)} + p^3(t)k$$

entonces

$$\dot{r} = \frac{\dot{p}^2}{m} + \frac{p^2\dot{m}}{m^2} + \dot{p}^3 k = \frac{\lambda_1}{m} + \frac{p^2}{m^2}(-k\alpha) + \left(\frac{p^2\alpha}{m^2}\right)k = \frac{\lambda_1}{m(t)}$$

Elegimos $\lambda_1 < 0$, tal que r sea decreciente. Calculamos

$$r(t^*) = 1 - \frac{\lambda_2 - \lambda_1 t^*}{m_0} + \lambda_3 k$$

y después ajustamos λ_2, λ_3 tal que $r(t^*) = 0$.

Por tanto, r es no creciente, $r(t^*) = 0$ y por tanto $r > 0$ en $[0, t^*), r < 0$ en $(t^*, \tau]$. Pero (M) establece que

$$\alpha(t) = \begin{cases} 1 & \text{si } r(t) < 0 \\ 0 & \text{si } r(t) > 0. \end{cases}$$

Así, el control óptimo cambia una única vez de 0 a 1 y, por tanto, nuestra teoría anterior sobre $\alpha(\cdot)$ satisface el principio del máximo de Pontryagin. ■

Conclusiones

A lo largo de este trabajo hemos visto cómo llegar a una red neuronal continua partiendo de una red neuronal discreta como es la ResNet. La conclusión es que podemos ver la formulación general de una red neuronal continua como un problema de minimización (optimización) paramétrica con restricciones dinámicas, ya que los parámetros de una red minimizan la función de pérdida para proporcionar los parámetros óptimos para la inferencia sobre nuevos datos.

En este camino vimos que si estas restricciones son la pertenencia a una curva, podemos servirnos de los multiplicadores de Lagrange para imponer dichas restricciones.

Si las restricciones pasan a ser de desigualdad, necesitaremos utilizar las condiciones de tipo KKT para la resolución del problema de minimización. Aun así, nos mantenemos en espacios de dimensión finita en los que las soluciones del sistema siguen siendo puntos de $\mathbb{R}^n$.

Cuando las restricciones son EDO que tienen que cumplirse en un intervalo de tiempo determinado, utilizamos el método del adjunto para definir los multiplicadores de Lagrange de nuestro problema, que pasan a denominarse coestados del sistema o adjuntos. La clave es que ahora son funciones y noo constantes. La función a minimizar para la determinación de los parámetros óptimos es un funcional de energía, que en última instancia será la función de pérdida de la red.

Para encontrar los ya mencionados coestados del sistema resolvemos la ecuación adjunta, que define la condición necesaria de optimalidad a través del principio del máximo de Pontryagin. Este principio generaliza las condiciones KKT para optimización con restricciones paramétricas y dinámicas. La técnica se basa en el cálculo del hamiltoniano del sistema, que nos va a permitir escribir las ecuaciones Euler-Lagrange del problema de optimización como un sistema de EDO de primer orden.

La optimización de los parámetros sujetos a restricciones dinámicas permnite así determinar la forma del control óptimo del sistema (los parámetros óptimos de la red) y habremos transformado una red discreta (Resnet) en un problema de control óptimo para EDO, es decir, la red continua ODENet. Estas redes tienen coste de memoria continuo, ya que no necesitamos almacenar ningún dato intermedio para la *backpropagation*. El coste de memoria suponía un gran cuello de botella en el entrenamiento de redes.

Además, dado que los *ODE solvers* actuales proporcionan garantías sobre el crecimiento del error de aproximación, monitorización del nivel de error y adaptan su estrategia de evaluación en tiempo real, el coste de evaluación del modelo escala

con la complejidad del problema. También se reduce el número de parámetros, ya que al parametrizar las dinámicas de la capa oculta como una función continua con respecto al tiempo, los parámetros de las capas cercanas se unen automáticamente.

Bibliografía

[1] T. Q. Chen, Y. Rubanova, J. Bettencourt, y D. Duvenaud, "Neural ordinary differential equations" *CoRR*, vol. abs/1806.07366, 2018. [Online]. Disponible en: http://arxiv.org/abs/1806.07366.

[2] K. He, X. Zhang, S. Ren, y J. Sun, "Deep residual learning for image recognition" en *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, págs. 770–778.

[3] M. E. Sander, P. Ablin, y G. Peyré, "Do residual neural networks discretize neural ordinary differential equations?" 2022. [Online]. Disponible en: https://arxiv.org/abs/2205.14612.

[4] J. Nocedal y S. J. Wright, *Numerical Optimization*, 2ª edición. Springer, 2006.

[5] E. Schiavi y I. Ramírez, "Apuntes de fundamentos matemáticos" 2020.

[6] S. Boyd y L. Vandenberghe, *Convex Optimization*, 7ª edición. Cambridge University Press, 2009.

[7] J. de Burgos, *Cálculo infinitesimal de varias variables*, 1ª edición. McGraw-Hill, 1995.

[8] S. Dineen, *Multivariate Calculus and Geometry*, 3ª edición. Springer, 2014.

[9] F. J. Martínez Sánchez, "El teorema de Karush-Kuhn-Tucker, una generalización del teorema de los multiplicadores de Lagrange, y programación convexa" *TEMat*, vol. 3, págs. 33–44, 2019. [Online]. Disponible en: https://temat.es/articulo/2019-p33.

[10] R. B. Israel, "A karush-kuhn-tucker example" 2006. [Online]. Disponible en: https://personal.math.ubc.ca/~israel/m340/kkt2.pdf.

[11] L. Bottou, "Large-scale machine learning with stochastic gradient descent" en *Proceedings of COMPSTAT'2010*, Y. Lechevallier y G. Saporta, (editores). Physica-Verlag HD, 2010, págs. 177–186.

[12] D. Kingma y J. Ba, "Adam: A method for stochastic optimization" *International Conference on Learning Representations*, 2014.

[13] L. Pontryagin, *Mathematical Theory of Optimal Processes*, ser. Classics of Soviet Mathematics. Taylor & Francis, 1987.

[14] Q. Li, L. Chen, C. Tai, y W. E, "Maximum principle based algorithms for deep learning" *Journal of Machine Learning Research*, vol. 18, no. 165, págs. 1–29, 2018. [Online]. Disponible en: http://jmlr.org/papers/v18/17-653.html.

[15] J. Aghili y O. Mula, "Depth-adaptive neural networks from the optimal control viewpoint" 2020. [Online]. Disponible en: https://arxiv.org/abs/2007.02428.

[16] X. Zhang, Z. Li, C. C. Loy, y D. Lin, "Polynet: A pursuit of structural diversity in very deep networks" *CoRR*, vol. abs/1611.05725, 2016. [Online]. Disponible en: http://arxiv.org/abs/1611.05725.

[17] G. Larsson, M. Maire, y G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals" *CoRR*, vol. abs/1605.07648, 2016. [Online]. Disponible en: http://arxiv.org/abs/1605.07648.

[18] A. M. Bradley, "Pde-constrained optimization and the adjoint method" 2019 (original 2010). [Online]. Disponible en: https://cs.stanford.edu/~ambrad/adjoint_tutorial.pdf.

[19] L. C. Evans, *An Introduction to Mathematical Optimal Control Theory.* Department of Mathematics, University of California, Berkeley, 2013.

[20] F. Gazzola y E. M. Marchini, "The moon lander optimal control problem revisited" *Mathematics in Engineering*, vol. 3, no. 5, págs. 1–14, 2021. [Online]. Disponible en: https://www.aimspress.com/article/doi/10.3934/mine.2021040.

Apéndices

A

Optimización

En la optimización se busca minimizar una función objetivo que depende de variables reales sin ningún tipo de restricción en los valores de dichas variables. La formulación matemática de este problema es

$$\min_{\mathbf{x}} f(\mathbf{x})$$

donde $\mathbf{x} \in \mathbb{R}^n$ es un vector real con $n \geq 1$ componentes y $f : \mathbb{R}^n \to \mathbb{R}$ es una función infinitamente diferenciable (de clase C^∞).

Normalmente carecemos de una perspectiva global de f. Solo conocemos los valores de f y quizá alguna derivada en un conjunto de puntos $x_1, x_2, x_3...$ Afortunadamente, los algoritmos tienden a elegir estos puntos y tratan de hacerlo de tal manera que identifican una solución de manera fiable utilizando poco tiempo y espacio. Aun así, la información sobre f no se calcula de manera sencilla, por lo que debemos decantarnos por algoritmos que no busquen esta información de manera innecesaria.

A.1. Conceptos básicos

Definición A.1.1 Un punto x^* es un **mínimo global** si existe un entorno $\mathcal{N}$ de x^* tal que $f(x^*) \leq f(x)$ para todo $x \in \mathcal{N}$.

Definición A.1.2 Un punto x^* es un **mínimo global estricto** si existe un entorno $\mathcal{N}$ de x^* tal que $f(x^*) < f(x)$ para todo $x \in \mathcal{N}$ con $x \neq x^*$.

Definición A.1.3 Un punto x^* es un **mínimo local aislado** si existe un entorno $\mathcal{N}$ de x^* tal que x^* es el único minimizador local en $\mathcal{N}$.

Si en las anteriores definiciones sustituimos $\leq$ por $\geq$ y $<$ por $>$, tenemos las definiciones de **máximo global**, **máximo global estricto** y **máximo local aislado**. En la siguiente figura podemos ver ejemplos de máximos y mínimos globales y locales:

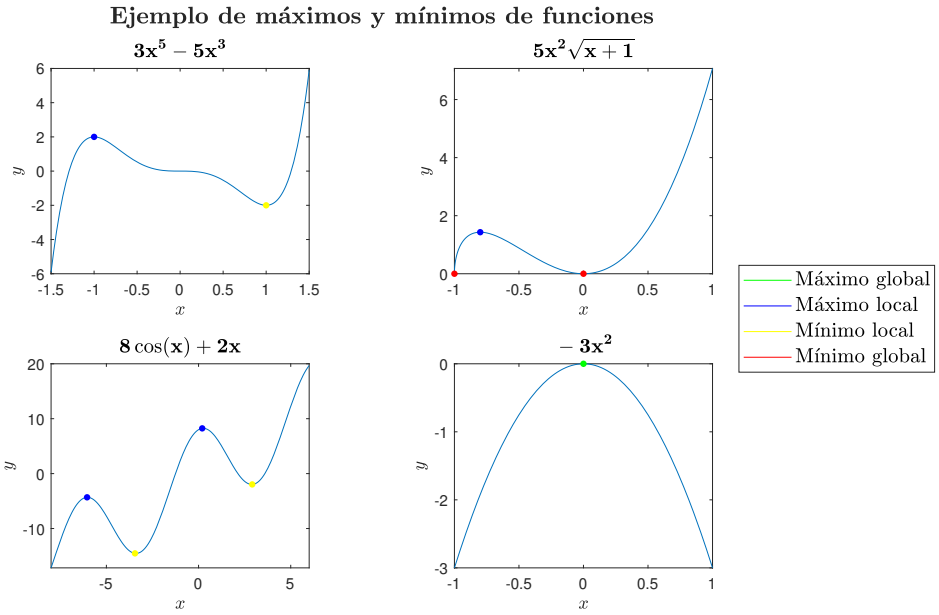

Figura A.1: Máximos y mínimos locales y aislados.

Algunos minimizadores locales estrictos no están aislados, como ocurre en la función

$$f(x) = x^4 \cos(1/x) + 2x^4, \quad f(0) = 0$$

que es dos veces continuamente diferenciable y tiene un minimizador local estricto en $x^* = 0$.

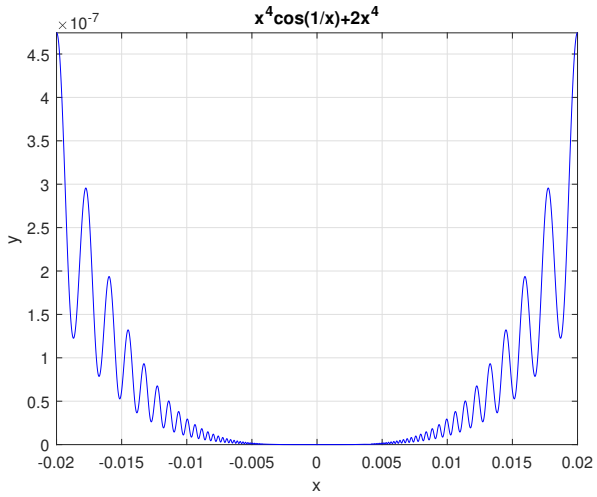

Figura A.2: Gráfica de la función de ejemplo.

Sin embargo, hay minimizadores locales estrictos en muchos puntos cercanos x_j, y podemos etiquetarlos de tal forma que $x_j \to 0$ según $j \to \infty$.

Mientras que los minimizadores locales estrictos no son siempre aislados, es cierto que todos los minimizadores locales aislados son siempre estrictos. En funciones como la del ejemplo,

es difícil encontrar el minimizador global porque los algoritmos se quedan atrapados en minimizadores locales. A veces tenemos información "global" adicional sobre f que puede ayudar a la hora de identificar mínimos globales. Un caso especial es el de las funciones convexas[1], para las cuales cada minimizador local lo es también global.

A.2. Condiciones de optimalidad

Teorema A.2.1 ***Teorema de Taylor***

Supongamos que $f : \mathbb{R}^n \to \mathbb{R}$ es una función continuamente diferenciable y que $\mathbf{p} \in \mathbb{R}^n$. Entonces tenemos que

$$f(\mathbf{x}+\mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x}+t\mathbf{p})^T\mathbf{p}$$

para algún $t \in (0,1)$. Más aún, si es doble y continuamente diferenciable, tenemos que

$$\nabla f(\mathbf{x}+\mathbf{p}) = \nabla f(\mathbf{x}) + \int_0^1 \nabla^2 f(\mathbf{x}+t\mathbf{p})\,\mathbf{p}\; dt \tag{A.1}$$

y que

$$f(\mathbf{x}+\mathbf{p}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T\mathbf{p} + \frac{1}{2}p^T\nabla^2 f(\mathbf{x}+t\mathbf{p})\,\mathbf{p}s$$

para algún $t \in (0,1)$.

Nota: La fórmula (A.1) se deduce como sigue:
Sea $g : \mathbb{R} \to \mathbb{R}^n$ tal que $g(t) = \nabla f(\mathbf{x}+t\mathbf{p})$. Tenemos que $g(0) = \nabla f(\mathbf{x})$ y $g(1) = \nabla f(\mathbf{x}+\mathbf{p})$. Por el teorema fundamental del cálculo

$$g(1) = g(0) + \int_0^1 g'(t)\,dt$$

Pero por la regla de la cadena sabemos que

$$g'(t) = \nabla^2 f(\mathbf{x}+t\mathbf{p})\mathbf{p}$$

y por tanto se deduce que

$$\nabla f(\mathbf{x}+\mathbf{p}) = \nabla f(\mathbf{x}) + \int_0^1 \nabla^2 f(\mathbf{x}+t\mathbf{p})\,\mathbf{p}\; dt$$

Las condiciones necesarias para la optimalidad se derivan de asumir que $\mathbf{x}^*$ es un minimizador local y probar después hechos sobre $\nabla f(\mathbf{x}^*)$ y $\nabla^2 f(\mathbf{x}^*)$.

Teorema A.2.2 ***Condiciones necesarias de primer orden***

Si $\mathbf{x}^*$ es un minimizador local y f es continuamente diferenciable (de clase $\mathcal{C}^1$) en un entorno abierto de $\mathbf{x}^*$, entonces $\nabla f(\mathbf{x}^*) = 0$.

[1] Aquellas funciones para las que se cumple que $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$ para todo x, y pertenecientes al dominio de la función y para todo $\lambda \in [0,1]$.

Demostración:
Supongamos por contradicción que $\nabla f(\mathbf{x}) \neq 0$. Definimos el vector $p = -\nabla f(\mathbf{x}^*)$ y vemos que $p^T \nabla f(\mathbf{x}^*) = -\|\nabla f(\mathbf{x}^*)\|^2 < 0$. Como ∇f es continua cerca de $\mathbf{x}^*$, existe un escalar $T > 0$ tal que

$$\mathbf{p}^T \nabla f(\mathbf{x}^* + t\mathbf{p}) > 0 \qquad \text{para todo } t \in [0, T]$$

Para algún $\bar{t} \in (0, T]$, tenemos por el teorema de Taylor que

$$f(\mathbf{x}^* + \bar{t}\mathbf{p}) = f(\mathbf{x}^*) + \bar{t}\mathbf{p}^T \nabla f(\mathbf{x}^* + t\mathbf{p}), \qquad \text{para algún } t \in (0, \bar{t})$$

Por tanto, $f(\mathbf{x}^* + \bar{t}\mathbf{p}) < f(\mathbf{x}^*)$ para todo $\bar{t} \in (0, T]$. Hemos encontrado una dirección que nos aleja de $\mathbf{x}^*$ en la que f decrece, por lo que $\mathbf{x}^*$ **no** es un minimizador local, lo que nos lleva a una contradicción.

■

Definición A.2.3 Decimos que $\mathbf{x}^*$ es un **punto estacionario** si $\nabla f(\mathbf{x}^*) = 0$. De acuerdo con el teorema A.2.2, cualquier minimizador global es un punto estacionario.

Para el siguiente resultado necesitamos recordar lo siguiente:

Definición A.2.4 La matriz cuadrada M se define como:

- **Definida positiva** si $\mathbf{x}^T M\mathbf{x} > 0$ para todo $\mathbf{x} \in \mathbb{R}^n$ y $\mathbf{x}^T M\mathbf{x} = 0 \leftrightarrow \mathbf{x} \equiv 0$. Todos sus autovalores son positivos.
- **Definida negativa** si $\mathbf{x}^T M\mathbf{x} < 0$ para todo $\mathbf{x} \in \mathbb{R}^n$ y $\mathbf{x}^T M\mathbf{x} = 0 \leftrightarrow \mathbf{x} \equiv 0$. Todos sus autovalores son negativos.
- **Semidefinida positiva** si $\mathbf{x}^T M\mathbf{x} \geq 0$ para todo $\mathbf{x} \in \mathbb{R}^n$. Todos sus autovalores son positivos o nulos.
- **Semidefinida negativa** si $\mathbf{x}^T M\mathbf{x} \leq 0$ para todo $\mathbf{x} \in \mathbb{R}^n$. Todos sus autovalores son negativos o nulos.

Teorema A.2.5 ***Condiciones necesarias de segundo orden***

Si $\mathbf{x}^*$ es un minimizador local de f y ∇^2 existe y es continua en un entorno abierto de $\mathbf{x}^*$, entonces $\nabla f(\mathbf{x}^*) = 0$ y $\nabla^2 f(\mathbf{x}^*)$ es semidefinida positiva.

Demostración:
Sabemos por el teorema A.2.2 que $\nabla f(\mathbf{x}^*) = 0$. Por reducción al absurdo, asumimos que $\nabla^2 f(\mathbf{x}^*)\mathbf{p} < 0$, y como $\nabla^2 f$ es continua en un entorno de $\mathbf{x}^*$, existe un escalar $T > 0$ tal que $\mathbf{p}^T \nabla^2 f(\mathbf{x}^* + t\mathbf{p})\,\mathbf{p} < 0$ para todo $t \in [0, T]$. Haciendo una expansión en serie de Taylor sobre $\mathbf{x}^*$, tenemos que para todo $\bar{t} \in (0, T]$ y para algún $t \in (0, \bar{t})$:

$$f(\mathbf{x}^* + \bar{t}\mathbf{p}) = f(\mathbf{x}^*) + \bar{t}\mathbf{p}^T \nabla f(\mathbf{x}^*) + \frac{1}{2}\bar{t}^2 \mathbf{p}^T \nabla^2 f(\mathbf{x}^* + t\mathbf{p})\,\mathbf{p} < f(\mathbf{x}^*)$$

Como ocurría con el A.2.2, hemos encontrado una dirección que nos aleja de $\mathbf{x}^*$ de manera dectreciente, por lo que, de nuevo, $\mathbf{x}^*$ no es un minimizador local.

■

Ahora vamos a describir las *condiciones suficientes*, que son condiciones sobre las derivadas de f en el punto $\mathbf{z}^*$ que garantizan que $\mathbf{x}^*$ es un minimizador local.

Definición A.2.6 Dada una función real f de n variables reales:

$$\begin{aligned} f : \mathbb{R}^n &\to \mathbb{R} \\ \mathbf{x} &\mapsto f(\mathbf{x}) \end{aligned}$$

Si existen todas las segundas derivadas parciales de f, se define la **matriz hessiana** de f como $H_f(\mathbf{x})$, donde

$$H_f(\mathbf{x})_{i,j} = \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}$$

es decir:

$$H(f) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} & \dfrac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Teorema A.2.7 ***Condiciones suficientes de segundo orden***

Sea $\nabla^2 f$ continua en un entorno abierto de $\mathbf{x}^*$ y sean $\nabla f(\mathbf{x}^*) = 0$ y $\nabla^2 f(\mathbf{x}^*)$ definida positiva. Entonces $\mathbf{x}^*$ es un minimizador local estricto de f.

Demostración:
Como la hessiana es continua y positiva definida en $\mathbf{x}^*$, podemos escoger un radio $r > 0$ tal que $\nabla^2 f(\mathbf{x}^*)$ permanece definida positiva para todo $\mathbf{x}$ en la bola abierta $\mathcal{D} = \{\mathbf{z} \mid \|\mathbf{z} - \mathbf{x}^*\| < r\}$. Tomando cualquier vector no nulo $\mathbf{p}$ con $\|\mathbf{p}\| < r$, tenemos $\mathbf{x}^* + \mathbf{p} \in \mathcal{D}$, por lo que

$$\begin{aligned} f(\mathbf{x}^* + \mathbf{p}) &= f(\mathbf{x}^*) + \mathbf{p}^T \nabla f(\mathbf{x}^*) + \tfrac{1}{2} \mathbf{p}^T \nabla^2 f(\mathbf{z})\, \mathbf{p} \\ &= f(\mathbf{x}^*) + \tfrac{1}{2} \mathbf{p}^T \nabla^2 f(\mathbf{z})\, \mathbf{p} \end{aligned}$$

siendo $\mathbf{z} = \mathbf{x}^* + t\mathbf{p}$ para algún $t \in (0, 1)$. Como $\mathbf{z} \in \mathcal{D}$, tenemos $\mathbf{p}^T \nabla^2 f(\mathbf{z})\, \mathbf{p} > 0$, y por tanto $f(\mathbf{x}^* + \mathbf{p}) > f(\mathbf{x}^*)$, obteniendo así el resultado esperado.

■

Es importante ver que las condiciones suficientes de segundo orden del teorema A.2.7 nos garantizan algo más fuerte que las condiciones necesarias anteriores: que el minimizador es estricto local. También puede verse que las condiciones suficientes de segundo orden no son realmente necesarias: Un punto $\mathbf{x}^*$ puede ser un minimizador local estricto y aun así no satisfacer las condiciones suficientes.

Un ejemplo de esto viene dado por la función $f(x) = x^4$, para la que el punto $x^* = 0$ es un minimizador local estricto en el que la matriz hessiana es la matriz nula (y, por tanto, no es definida positiva).

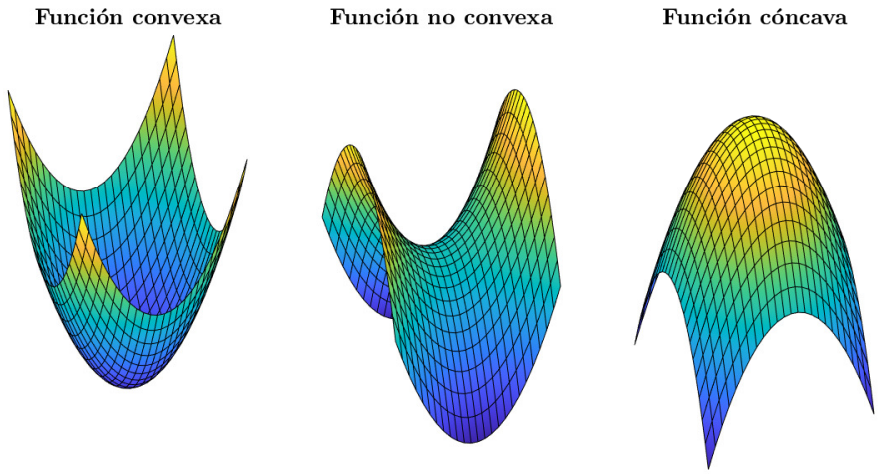

Figura A.3: Ejemplos de concavidad y convexidad de funciones.

Teorema A.2.8 *Cuando f es convexa, cualquier minimizador local $\mathbf{x}^*$ es un minimizador global de f. Además, si f es diferenciable, entonces cualquier punto estacionario $\mathbf{x}^*$ es un minimizador global de f.*

Demostración:
Supongamos que $\mathbf{x}^*$ es un minimizador local pero no global. Entonces podemos encontrar un punto $\mathbf{z} \in \mathbb{R}^n$ con $f(\mathbf{z}) < f(\mathbf{x}^*)$. Consideremos el segmento que une $\mathbf{x}^*$ con $\mathbf{z}$, es decir

$$\mathbf{x} = \lambda\,\mathbf{z} + (1-\lambda)\,\mathbf{x}^* \qquad \text{para algún } \lambda \in (0,1] \tag{A.2}$$

Por la convexidad de f tenemos

$$f(\mathbf{x}) \leq \lambda f(\mathbf{z}) + (1-\lambda) f(\mathbf{x}^*) < f(\mathbf{x}^*) \tag{A.3}$$

Cualquier entorno $\mathcal{N}$ de $\mathbf{x}^*$ contiene una parte del segmento definido en (A.2), por lo que siempre habrá puntos $\mathbf{x} \in \mathcal{N}$ que satisfacen (A.3). Por tanto, $\mathbf{x}^*$ no es un minimizador local.

Para la segunda parte del teorema, supongamos que $\mathbf{x}^*$ no es un minimizador global y elegimos un $\mathbf{z}$ como el anterior. Entonces, por convexidad, tenemos

$$\nabla f(\mathbf{x}^*)^T(\mathbf{z}-\mathbf{x}^*) = \frac{d}{d\lambda} f(\mathbf{x}^* + \lambda(\mathbf{z}-\mathbf{x}^*))|_{\lambda=0}$$

$$= \lim_{\lambda \to 0} \frac{f(\mathbf{x}^* + \lambda(\mathbf{z}-\mathbf{x}^*)) - f(\mathbf{x}^*)}{\lambda}$$

$$\leq \lim_{\lambda \to 0} \frac{\lambda f(\mathbf{z}) + (1-\lambda) f(\mathbf{x}^*) - f(\mathbf{x}^*)}{\lambda}$$

$$= f(\mathbf{z}) - f(\mathbf{x}^*) < 0$$

Por tanto, $\nabla f(\mathbf{x}^*) \neq 0$, así que $\mathbf{x}^*$ no es un punto estacionario.

■

B

Ejemplos para ilustrar la teoría

Ejemplo básico de minimización con restricciones

Ejemplo B.0.1

$$\text{mín } f(x,y), \quad \text{sujeto a } g(x,y)=0,$$

siendo

$$f(x,y)=(y+100)^2+0.01x^2, \qquad g(x,y)=y-\cos(x)$$

Sin la restricción, que es pertenecer a la curva de nivel $g(x,y)=0$, el problema tiene la solución única $x=0$, $y=-100$. Con la restricción, hay soluciones locales cerca de los puntos

$$\mathbf{p}^{(k)}=(k\pi,-1)^T, \quad \text{para } k=\pm1,\pm3,\pm5,....$$

ya que la función a minimizar es, sustituyendo y por $\cos(x)$:

$$f(x,y)|_{y=\cos(x)}=f(x,\cos(x))=f(x)=(\cos(x)+100)^2+0.01x^2$$

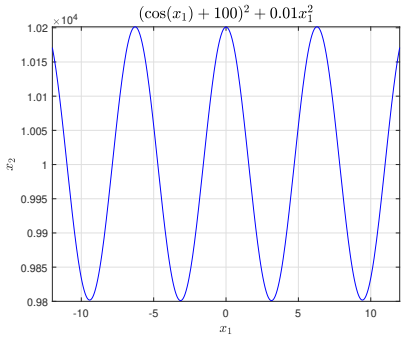

Figura B.1: Gráfica de la función de ejemplo.

Las definiciones de los distintos tipos de soluciones locales son extensiones de las definiciones correspondientes para el caso sin restricciones, excepto que ahora consideramos puntos factibles a los entornos de $\mathbf{x}^*$.

Ejemplo para ilustrar la formulación del epígrafe

Ejemplo B.0.2

$$\min_{x\in\mathbb{R}} f(x), \qquad \text{siendo } f(x) = \max_{x\in\mathbb{R}}(x^2, x)$$

que, como vemos en la figura B.2, tiene como puntos críticos $x = 0$ y $x = 1$, y como solución $x^* = 0$. Los puntos críticos se han determinado por ser puntos de no derivabilidad. Obsérvese que aparecen en la figura B.2 como puntos angulosos (derivadas laterales distintas). El problema B.0.2 es un problema de optimización no regular sin restricciones. La no regularidad radica en la no derivabilidad de la función máximo.

La función máximo de funciones continuas es continua, pero no es derivable aunque las funciones lo sean. Por esta falta de diferenciabilidad, en inteligencia artificial se utiliza una función (de activación) llamada *softmax*, que representa una aproximación diferenciable de la función máximo.

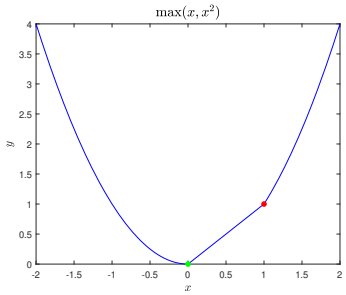

Figura B.2: Gráfica de la función $f(x) = \max_{x\in\mathbb{R}}(x, x^2)$.

La función a minimizar puede expresarse como una función descrita a trozos:

$$f(x) = \max_{x\in\mathbb{R}}(x^2, x) = \begin{cases} x^2 & \text{si} \quad x \leq 0 \\ x & \text{si} \quad 0 < x \leq 1 \\ x^2 & \text{si} \quad x > 1 \end{cases}$$

Como puede demostrarse mediante el cálculo de los límites en los puntos críticos $x = 0$ y $x = 1$ y como se aprecia en la figura B.2, la función $f(x)$ es continua en su dominio $\mathcal{D}(f) \equiv \mathbb{R}$. Es decir, $\forall x \in \mathcal{D}(f)$, tenemos que para toda sucesión $\{x_n\}$ de puntos de $\mathcal{D}(f)$ tales que $x_n \to x$, se tiene que $f(x_n) \xrightarrow{f} (x)$.

Calculando los límites en los extremos del dominio:

$$\lim_{x \to -\infty} f(x) = \lim_{x \to +\infty} f(x) = +\infty$$

Por el teorema de Weierstrass generalizado existe al menos un punto de mínimo absoluto de la función en $\mathbb{R}$.

Calculando la primera derivada de $f(x)$ tenemos:

$$f'(x) = \begin{cases} 2x & \text{si} \quad x < 0 \\ 1 & \text{si} \quad 0 < x < 1 \\ 2x & \text{si} \quad x > 1 \end{cases}$$

y se observa la falta de derivabilidad en los puntos críticos $x = 0$ y $x = 1$.

Sin embargo, en hipótesis de continuidad podemos utilizar las condiciones suficientes de primer orden para la clasificación de extremos relativos (criterio de la derivada primera) basadas en los cambios de monotonía en un entorno con agujero de los puntos críticos. En el intervalo $(-\infty, 0)$ la función es decreciente y en el intervalo $(0, 1)$ la función es creciente; luego el punto $(0, 0)$ es un mínimo relativo de la función $f(x)$. En el otro punto crítico no hay cambio de monotonía. Como no hay otros puntos críticos, podemos asegurar que el punto $(0, 0)$ es el mínimo absoluto de la función $f(x)$ y, por tanto, el punto solución de nuestro problema.

A pesar de haber resuelto el problema en su formulación no regular, es posible obtener una formulación regular del problema añadiendo una variable artificial t y escribiendo

$$\text{mín } t \qquad \text{sujeto a} \quad t \geq x, \quad t \geq x^2$$

La función a minimizar, $f(t) = t$ sigue siendo continua (es una función lineal) y, además, es regular. Las restricciones derivan de la función $f(x)$ anterior (t tiene que ser mayor que x y que x^2) y definen un conjunto factible convexo y cerrado. La función es inferiormente acotada, ya que ambas desigualdades se tienen que cumplir, así que sabemos que $t \geq 0$.

Una representación gráfica del problema se muestra en la figura siguiente:

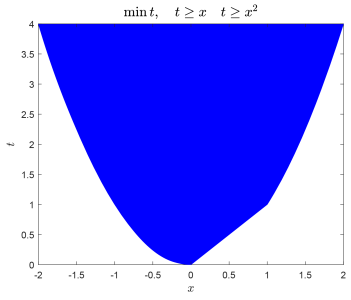

Figura B.3: Gráfica del problema mín t sujeto a $t \geq x, t \geq x^2$.

El valor mínimo $t = 0$ se alcanza para $x = 0$. Puede verse fácilmente en la figura que el punto $(0, 0)$ es el punto perteneciente a la región factible (coloreada en azul) en el que alcanzamos el mínimo.

Ejemplo simple para la ilustración del método del adjunto

Ejemplo B.0.3 Siguiendo el procedimiento expuesto en Pseudocódigo del algoritmo del cálculo del gradiente (método del adjunto) vamos a calcular el gradiente de

$$\int_0^T x\, dt, \quad \text{sujeto a} \quad \begin{cases} \dot{x} = bx \\ x(0) - a = 0. \end{cases}$$

En este caso las dimensiones del problema son $n = 1$, $d = 2$, ya que tenemos una única EDO y 2 parámetros, a y b. Luego $\mathbf{x} = x \in \mathbb{R}$, $x = x(t)$ y $\mathbf{p} = [p_1, p_2] = [a, b]^T \in \mathbb{R}^2$. Se tiene

$$f(x, \mathbf{p}, t) = x, \qquad h(x, \dot{x}, \mathbf{p}, t) = \dot{x} - bx, \qquad g(x(0), \mathbf{p}) = x(0) - a.$$

Definimos mediante (3.7) el lagrangiano aumentado para el problema de minimización sin restricciones para la variable $x(t)$ y los multiplicadores de Lagrange $\lambda(t)$ y μ.

$$\mathcal{L}(x, \mathbf{p}) \doteq \int_0^T [f(x, \mathbf{p}, t) + \lambda^T h(x, \dot{x}, \mathbf{p}, t)]\, dt + \mu^T g(x(0), \mathbf{p}) =$$

$$= \int_0^T [x + \lambda^T (\dot{x} - bx)]\, dt + \mu^T (x(0) - a)$$

Siguiendo los pasos indicados

1. Resolución del PVI. La EDO que define este problema es lineal, de primer orden y separable, por lo que su resolución es directa. Al integrar la EDO obtenemos

   $$x(t) = ae^{bt}$$

   Utilizando el cálculo simbólico de Matlab podemos calcular explícitamente y de forma paramétrica la solución del PVI. Para ello se definen las variables simbólicas del problema y el PVI. A continuación se utiliza el comando dsolve.m mediante el siguiente código

Código B.1: Primera parte del algoritmo para calcular $d_{\mathbf{p}}F$

```
% Autores: Laura Medina Henche y Emanuele Schiavi
% Universidad Rey Juan Carlos, 2024
% El presente software se distribuye segun la licencia
% GPLv3 (https://www.gnu.org/licenses/gpl-3.0.en.html)

syms a b t  x x0  reals
syms x(t)
ODE = diff(x, t) == b.*x
x0 = a
sol(t) = dsolve (ODE, x(0) == x0)
sol_p(a,b,t) = sol(t)
```

2. A partir de las ecuaciones

$$f(x,\mathbf{p},t)=x, \qquad h(x,\dot{x},\mathbf{p},t)=\dot{x}-bx$$

calculamos las derivadas parciales

$$\partial_x f=1, \qquad \partial_x h=-b, \qquad \partial_{\dot{x}} h=1$$

que nos permiten calcular la EDO adjunta

$$\partial_x f+\lambda^T\left(\partial_x h-d_t\,\partial_{\dot{x}}h\right)-\dot{\lambda}\,\partial_{\dot{x}}h=0$$

complementada con la condición final $\lambda(T)=0$. Sustituyendo las expresiones calculadas se tiene el problema de valores finales (PVF)

$$\begin{cases} 1-b\lambda-\dot{\lambda}=0 \\ \lambda(T)=0 \end{cases}$$

La EDO adjunta es lineal y el problema se puede resolver mediante el método de variación de las constantes obteniendo la solución o coestado

$$\lambda(t)=\frac{1}{b}\left(1-e^{b\,(T-t)}\right)$$

El código en matlab que resuelve este segundo paso es, agregado al código anterior, el siguiente:

Código B.2: Segunda parte del algoritmo para calcular $d_{\mathbf{p}}F$

```
% Autores: Laura Medina Henche y Emanuele Schiavi
% Universidad Rey Juan Carlos, 2024
% El presente software se distribuye segun la licencia
% GPLv3 (https://www.gnu.org/licenses/gpl-3.0.en.html)

syms a b t x dx reals
syms  lambda(t)
p=[a;b];
f = x  % f(x,p,t) integrando
dfx = diff(f,x)  % parcial x
%% restriccion EDO explicita
h_bar = b*x  %h_bar(x,p,t)
```

```
dh_bar = diff(h_bar,x)
%% implicita h=0
h = dx-h_bar  %h(x,dx,p,t)
dhx = diff(h,x)
dhp = [diff(h,a);diff(h,b)]
dhdx = diff(h,dx)
%% EDO para el ADJUNTO
% lambda(t)=multiplicador para la EDO
EDO_adj = ...
    dfx+lambda*(dhx-diff(dhdx,t))-diff(lambda,t)*dhdx
%% resolucion simbolica
EDO_a = diff(lambda, t) ==1- b.*lambda
% Condicion final:
lambda_T = 0
% Resolvemos
lam(t) = dsolve (EDO_a, lambda(T) == lambda_T)
lam_p(a,b,t) = lam(t)
```

3. $\partial_p f = [0 \;\; 0]$, $\partial_p h = [0 \;\; -x]$, $g_{x(0)} = 1$ y $g_p = [-1 \;\; 0]$. Calculamos el gradiente de F como sigue

$$d_{\mathbf{p}} F = \int_0^T \left[f_{\mathbf{p}} + \lambda^T \partial_{\mathbf{p}} h \right] dt + \lambda^T \partial_{\dot{x}} h \big|_{t=0} \, g_{x(0)}^{-1} \, g_{\mathbf{p}} =$$

$$= \int_0^T (0 \quad 0) + \lambda(t)(0 \quad -x) \, dt + \lambda(0) \cdot 1 \cdot 1^{-1} \cdot (-1 \quad 0) =$$

$$= \left(b^{-1} \left(-1 + e^{b\,T} \right) \qquad \frac{a}{b} \, T \, e^{b\,T} - \frac{a}{b^2} \left(e^{b\,T} - 1 \right) \right)$$

Es decir

$$d_a F = b^{-1} \left(-1 + e^{b\,T} \right)$$

$$d_b F = \frac{a}{b} \, T \, e^{b\,T} - \frac{a}{b^2} \left(e^{b\,T} - 1 \right)$$

El código en matlab que resuelve este último paso es:

Código B.3: Tercera parte del algoritmo para calcular $d_{\mathbf{p}} F$

```
% Autores: Laura Medina Henche y Emanuele Schiavi
% Universidad Rey Juan Carlos, 2024
% El presente software se distribuye segun la licencia
% GPLv3 (https://www.gnu.org/licenses/gpl-3.0.en.html)

p = [a;b];
dfp = [diff(f,a);diff(f,b)]
%%
h_bar = b*x  %h_bar(x,p,t)
h = dx-h_bar  %h(x,dx,p,t)
dhx = diff(h,x)
dhp = [diff(h,a);diff(h,b)]
dhdx = diff(h,dx)
%% restriccion CI g=0
g = x0-a  % g(x0,p)=g(x0,a,b)
dgx0 = diff(g,x0)
dgp = [diff(g,a);diff(g,b)]
%% variable para cancelar un termino del gradiente
```

```
% multiplicadores asociado a cond. iniciales
muu = lam(0)*dhdx/dgx0
% dhdx =1 luego no evaluo.
%%
integrando = dfp+lam(t).*dhp % verifica: es -lam*x
%% termino integral
comp_1 = int(integrando,0,T)
%%
comp_2 = muu*dgp
%% Gradiente de F en p
dFp = comp_1+comp_2
```

Los siguientes gráficos explicativos se han obtenido mediante el código de python que puede encontrarse en el anexo.

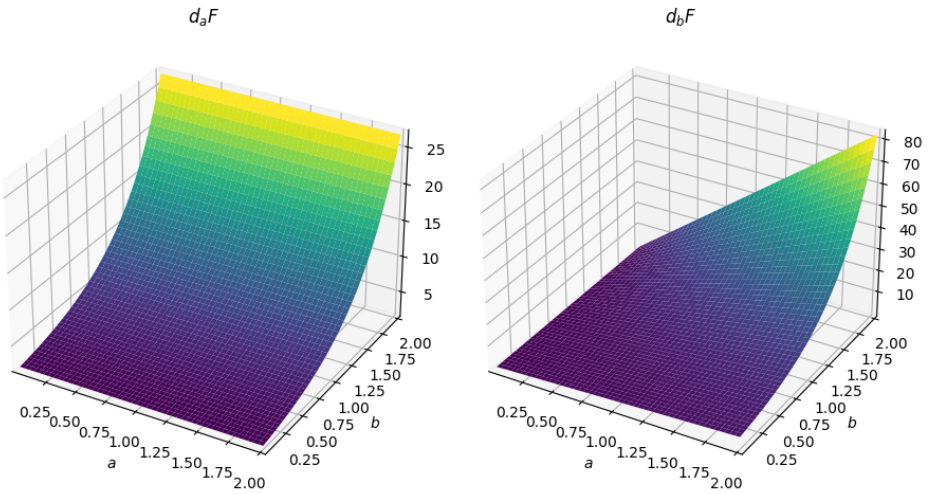

Figura B.4: Gráficos del cálculo del gradiente de $\int_0^T x\,dt$ (Ejemplo B.0.3). Se ha representado cada componente del gradiente respecto a sus variables, siempre considerando $T = 2$. Podemos ver que para b cercanas a 0, d_aF se mantiene estable y cercano a 0, pero en cuanto aumentamos el valor de b, el valor del gradiente aumenta considerablemente. En cuanto a d_bF, para valores de a o b cercanos a 0 el gradiente se mantiene constante y cercano a 0, pero se dispara cuando aumentamos alguno de los dos valores.

C

Material complementario al capítulo 1

C.1. Definiciones

Definición C.1.1 Dado un punto factible $\mathbf{x}$, decimos que $\{z_k\}$ es una **secuencia factible** tendente a $\mathbf{x}$ si $z_k \in \Omega$ para todo k suficientemente grande y $\{z_k\} \to \mathbf{x}$. Una **tangente** es una dirección limitante de la secuencia factible.

Definición C.1.2 El vector $\mathbf{d}$ se denomina **vector tangente** a Ω en un punto $\mathbf{x}$ si hay una secuencia factible $\{z_k\}$ que tiende a $\mathbf{x}$ y una secuencia de escalares positivos $\{t_k\}$ con $\{t_k\} \to 0$ tal que

$$\lim_{k \to \infty} \frac{z_k - x}{t_k} = d \tag{C.1}$$

El conjunto de todas las tangentes a Ω en $\mathbf{x}^*$ se denomina **cono tangente** y se denota $T_\Omega(\mathbf{x}^*)$.

Definición C.1.3 Dado un punto factible $\mathbf{x}$ y el conjunto activo $\mathcal{A}(\mathbf{x})$, el **conjunto de direcciones factibles linealizadas** $\mathcal{F}(\mathbf{x})$ es

$$\mathcal{F}(\mathbf{x}) = \left\{ d \mid \begin{array}{ll} d^T \nabla c_i(\mathbf{x}) = 0, & \text{para todo } i \in \mathcal{E} \\ d^T \nabla c_i(\mathbf{x}) \geq 0, & \text{para todo } i \in \mathcal{A}(x) \cap \mathcal{I} \end{array} \right\}$$

Es importante ver que la definición de cono tangente no depende de la especificación algebraica del conjunto Ω, solo de su geometría. Sin embargo, el conjunto factible de direcciones linealizadas sí que depende de la definición de las restricciones c_i, $i \in \mathcal{E} \cup \mathcal{I}$.

Definición C.1.4 Las **cualificaciones de restricciones** de primer orden son propiedades de la descripción analítica de un conjunto que aseguran que su estructura en un entorno de un punto factible dado puede ser construido mediante aproximaciones (de primer orden) de las funciones de restricción que definen el conjunto.

En otras palabras, son condiciones bajo las cuales el conjunto de direcciones factibles linealizadas $\mathcal{F}(\mathbf{x})$ es similar al cono tangente $T_\Omega(\mathbf{x})$. De hecho, la mayoría de cualificaciones de restricciones aseguran que los dos conjuntos son iguales.

C.2. Relación entre el cono tangente y el conjunto de direcciones factibles

El siguiente resultado utiliza una cualificación de restricciones (*CRIL*) para relacionar el cono tangente con el conjunto $\mathcal{F}$ de direcciones factibles. En la siguiente prueba y en los resultados posteriores vamos a utilizar la notación $A(\mathbf{x}^*)$ para representar la matriz cuyas filas son los gradientes de las restricciones activas en el punto óptimo, es decir

$$A(\mathbf{x}^*)^T = [\nabla c_i(\mathbf{x}^*)]_{i\in\mathcal{A}(\mathbf{x}^*)}$$

donde el conjunto activo $\mathcal{A}(\mathbf{x}^*)$ se define como en 1.3.1.

Lema C.2.1 *Sea $\mathbf{x}^*$ un punto factible. Entonces se cumple*

I $T_\Omega(\mathbf{x}^*) \subset \mathcal{F}(\mathbf{x}^*)$

II Si la CRIL se verifica en $\mathbf{x}^*$, entonces $\mathcal{F}(\mathbf{x}^*) = T_\Omega(\mathbf{x}^*)$

Demostración:
Supongamos sin pérdida de generalidad que todas las restricciones $c_i(\cdot), i = 1, 2, ..., m$, son activas en $\mathbf{x}^*$ (podemos hacer esto ordenando y renombrando las restricciones activas e ignorando las inactivas, que son irrelevantes en algún entorno de $\mathbf{x}^*$).

Para probar I, sean x_k y t_k las secuencias para las cuales se satisface (C.1), es decir

$$\lim_{k\to\infty} \frac{z_k - \mathbf{x}}{t_k} = d$$

Además, $t_k > 0$ para todo k. De esta definición, tenemos que

$$z_k = \mathbf{x}^* + t_k d + o(t_k) \tag{C.2}$$

Tomando $i \in \mathcal{E}$ y utilizando el teorema de Taylor, tenemos que

$$\begin{aligned} 0 &= \frac{1}{t_k} c_i\left(z_k\right) \\ &= \frac{1}{t_k}\left[c_i\left(\mathbf{x}^*\right) + t_k \nabla c_i\left(\mathbf{x}^*\right)^T d + o\left(t_k\right)\right] \\ &= \nabla c_i\left(\mathbf{x}^*\right)^T d + \frac{o\left(t_k\right)}{t_k} \end{aligned}$$

Tomando el límite cuando $k \to \infty$, el último término de la expresión desaparece y tenemos $\nabla c_i\left(\mathbf{x}^*\right)^T d = 0$, como queríamos.

Para las restricciones de desigualdad activas $i \in \mathcal{A} \cap \mathcal{I}$, tenemos también que

$$\begin{aligned} 0 &\leq \frac{1}{t_k} c_i\left(z_k\right) \\ &= \frac{1}{t_k}\left[c_i\left(\mathbf{x}^*\right) + t_k \nabla c_i\left(\mathbf{x}^*\right)^T d + o\left(t_k\right)\right] \\ &= \nabla c_i\left(\mathbf{x}^*\right)^T d + \frac{o\left(t_k\right)}{t_k} \end{aligned}$$

Por tanto, como sucedía con las restricciones de igualdad, $\nabla c_i\left(\mathbf{x}^*\right)^T d = 0$.

Para II, usamos el teorema de la función implícita. En primer lugar, como la CRIL se verifica, tenemos por la definición 1.5.1 que la matriz $A(\mathbf{x}^*)_{m\times n}$ tiene rango completo $= m$. Sea Z una matriz cuyas columnas son una base para el kernel de $A(\mathbf{x}^*)$; es decir

$$Z \in \mathbb{R}^{n\times(n-m)}, \qquad Z \text{ tiene rango completo} = n, \qquad A(\mathbf{x}^*)Z = 0$$

Elegimos $d \in \mathcal{F}(\mathbf{x}^*)$ de manera arbitraria, y suponemos que $\{t_k\}_{k=0}^{\infty}$ es cualquier serie de escalares positivos tal que $lim_{k\to\infty} t_k = 0$. Definimos el sistema de ecuaciones parametrizado $R : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ como

$$R(z,t) = \begin{bmatrix} c(z) - tA\left(\mathbf{x}^*\right) d \\ Z^T\left(z - \mathbf{x}^* - td\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{C.3}$$

Tenemos que las soluciones $z = z_k$ de este sistema para $t = t_k > 0$ pequeñas dan una secuencia factible que se aproxima a $\mathbf{x}^*$ y satisface (C.1).
En $t = 0$, $z = \mathbf{x}^*$ y el jacobiano de R en este punto es

$$\nabla_z R(\mathbf{x}^*, 0) = \begin{bmatrix} A(\mathbf{x}^*) \\ Z^T \end{bmatrix}$$

que es no singular por construcción de Z. Por tanto, de acuerdo con el teorema de la función implícita, el sistema (C.3) tiene una solución z_k para todos los valores de t_k lo suficientemente pequeños. Además, tenemos de (C.3) y la definición C.1.3 que

$$i \in \mathcal{E} \Rightarrow c_i\left(z_k\right) = t_k \nabla c_i\left(\mathbf{x}^*\right)^T d = 0$$

$$i \in \mathcal{A}\left(\mathbf{x}^*\right) \cap \mathcal{I} \Rightarrow c_i\left(z_k\right) = t_k \nabla c_i\left(\mathbf{x}^*\right)^T d \geq 0$$

por lo que z_k es factible.

Queda por probar que (C.1) se cumple para la $\{z_k\}$ elegida. Utilizando el hecho de que $R(z_k, t_k) = 0$ para todo k, junto con el teorema de Taylor, tenemos

$$\begin{aligned} 0 = R\left(z_k, t_k\right) &= \begin{bmatrix} c\left(z_k\right) - t_k A\left(\mathbf{x}^*\right) d \\ Z^T\left(z_k - x^* - t_k d\right) \end{bmatrix} \\ &= \begin{bmatrix} A\left(\mathbf{x}^*\right)\left(z_k - \mathbf{x}^*\right) + o\left(\left\|z_k - \mathbf{x}^*\right\|\right) - t_k A\left(\mathbf{x}^*\right) d \\ z^T\left(z_k - \mathbf{x}^* - t_k d\right) \end{bmatrix} \\ &= \begin{bmatrix} A\left(\mathbf{x}^*\right) \\ Z^T \end{bmatrix}\left(2k - \mathbf{x}^* - t_k d\right) + o\left(\left\|z_k - \mathbf{x}^*\right\|\right) \end{aligned}$$

Dividiendo esta expresión por t_k y usando la no singularidad de la matriz de coeficientes del primer término, tenemos

$$\frac{z_k - \mathbf{x}^*}{t_k} = d + o\left(\frac{\|z_k - \mathbf{x}^*\|}{t_k}\right)$$

de lo que se deduce que (C.1) se satisface para $\mathbf{x} = \mathbf{x}^*$. Por tanto, $d \in T_\Omega(\mathbf{x}^*)$ para un $d \in \mathcal{F}(\mathbf{x}^*)$ arbitrario, por lo que la prueba de II está completa.

■

Como se ha mencionado antes, una solución local de (1.1) es un punto x en el que todas las secuencias factibles tienen la propiedad de que $f(z_k) \geq f(\mathbf{x})$ para todo k suficientemente largo. El siguiente resultado muestra que si dicha secuencia existe, entonces sus direcciones limitantes tienen que hacer producto interior con el gradiente de la función objetivo no negativo.

Teorema C.2.2 *Si $\mathbf{x}^*$ es una solución local de 1.1, entonces tenemos*

$$\nabla f(\mathbf{x}^*)^T d \geq 0, \quad \textit{para todo } d \in T_\Omega(\mathbf{x}^*)$$

Demostración:
Supongamos por contradicción que existe una tangente d para la cual $\nabla f(\mathbf{x}^*)^T d < 0$. Sean $\{z_k\}$ y $\{t_k\}$ las secuencias que satisfacen la definición C.1.2 para esta d.

Tenemos que

$$\begin{aligned} f(z_k) &= f(\mathbf{x}^*) + (z_k - \mathbf{x}^*)^T \nabla f(\mathbf{x}^*) + o(\|z_k - \mathbf{x}^*\|) \\ &= f(\mathbf{x}^*) + t_k d^T \nabla f(\mathbf{x}^*) + o(t_k), \end{aligned}$$

donde la segunda línea se sigue de (C.2). Como $d^T \nabla f(\mathbf{x}^*) < 0$, el término restante es dominado por el término de primer orden, es decir

$$f(z_k) < f(\mathbf{x}^*) + \frac{1}{2} t_k d^T \nabla f(\mathbf{x}^*)$$

para todo k lo suficientemente grande.

Por tanto, dado cualquier entorno de $\mathbf{x}^*$, podemos elegir un k lo suficientemente grande para el cual z_k esté en dicho entorno y tenga un valor más bajo para la función objetivo f. Por tanto, $\mathbf{x}^*$ no es una solución local.

■

C.3. Lema de Farkas

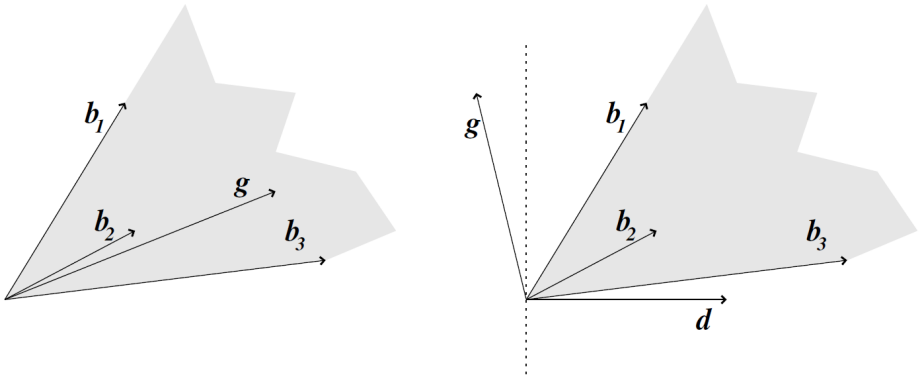

Figura C.1: Lema de Farkas.

Lema C.3.1 ***Lema de Farkas***

Consideremos un cono K definido como sigue:

$$K = \{By + Cw \mid y \geq 0\}$$

donde B y C son matrices de dimension $n \times m$ y $n \times p$ respectivamente, e $\mathbf{y}$ y $\mathbf{w}$ son vectores de dimensiones apropiadas. Dado un vector $\mathbf{g} \in \mathbb{R}^n$, el lema de Farkas establece que solo uno de los siguientes enunciados es cierto. O bien $\mathbf{g} \in K$, o bien existe un vector $\mathbf{d} \in \mathbb{R}^n$ tal que

$$\mathbf{g}^T\mathbf{d} < 0, \quad B^T\mathbf{d} \geq 0, \quad C^T\mathbf{d} = 0 \tag{C.4}$$

Estos dos casos están ilustrados en la figura C.3 para el caso de B de tres columnas, C nula y $n = 2$. En el segundo caso, el vector $\mathbf{d}$ define un hiperplano separador, que es un plano en $\mathbb{R}^n$ que separa al vector $\mathbf{g}$ del cono K.

Demostración:
Primero vamos a demostrar que las dos alternativas no pueden cumplirse simultáneamente. Si $\mathbf{g} \in K$, existen vectores $\mathbf{y} \geq 0$ y $\mathbf{w}$ tal que $\mathbf{g} = B\mathbf{y} + C\mathbf{w}$. Si también existe un $\mathbf{d}$ que cumple (C.4), tenemos por productos interiores que

$$0 > \mathbf{d}^T\mathbf{g} = \mathbf{d}^T B\mathbf{y} + \mathbf{d}^T C\mathbf{w} = \left(B^T\mathbf{d}\right)^T \mathbf{y} + \left(C^T\mathbf{d}\right)^T \mathbf{w} \geq 0$$

de donde la última desigualdad se sigue de $C^T\mathbf{d} = 0$, $B^T\mathbf{d} \geq 0$ e $\mathbf{y} \geq 0$. Por tanto, las dos alternativas no pueden darse al mismo tiempo. Ahora vamos a probar que una de las alternativas se cumple. Vamos a construir $\mathbf{d}$ con las propiedades (C.4) cuando $\mathbf{g} \notin K$. Necesitamos usar el hecho de que K es un conjunto cerrado. Sea $\hat{\mathbf{s}}$ el vector de K más próximo a $\mathbf{g}$ en el sentido de la norma euclídea. Como K es cerrado, $\hat{\mathbf{s}}$ está bien definido y viene dado por la solución del problema de optimización:

$$\text{mín } \|\mathbf{s} - \mathbf{g}\|_2^2 \qquad \text{sujeto a } \mathbf{s} \in K$$

Dado que $\hat{\mathbf{s}} \in K$ y K es un cono, $\alpha\hat{\mathbf{s}} \in K$ para todo escalar $\alpha \geq 0$. Como $\|\alpha\hat{\mathbf{s}}\|_2^2$ es mínimo en $\alpha = 1$, tenemos que:

$$\begin{aligned} \frac{d}{d\alpha}\|\alpha\hat{\mathbf{s}} - \mathbf{g}\|2^2\Big|_{\alpha=1} = 0 \Rightarrow \left(-2\hat{\mathbf{s}}^T\mathbf{g} + 2t\hat{\mathbf{s}}^T\hat{\mathbf{s}}\right)\big|_{\alpha=1} = 0 \\ \Rightarrow \hat{\mathbf{s}}^T(\hat{\mathbf{s}} - g) = 0 \end{aligned} \tag{C.5}$$

Ahora sea $\mathbf{s}$ cualquier otro vector en K. Como K es convexo, tenemos por la propiedad minimizante de $\hat{\mathbf{s}}$ que:

$$\|\hat{\mathbf{s}} + \theta(\mathbf{s} - \hat{\mathbf{s}}) - \mathbf{g}\|_2^2 \geq \|\hat{\mathbf{s}} - \mathbf{g}\|_2^2 \quad \text{para todo } \theta \in [0, 1]$$

y por tanto

$$2\theta(\mathbf{s} - \hat{\mathbf{s}})^T(\hat{\mathbf{s}} - \mathbf{g}) + \theta^2\|\mathbf{s} - \hat{\mathbf{s}}\|_2^2 \geq 0$$

Dividiendo esta expresión por θ y tomando el límite cuando $\theta \to 0$, tenemos $(\mathbf{s} - \hat{\mathbf{s}})^T(\hat{\mathbf{s}} - \mathbf{g}) \geq 0$. Entonces, por (C.5),

$$\mathbf{s}^T(\hat{\mathbf{s}} - \mathbf{g}) \geq 0, \quad \text{para todo } \mathbf{s} \in K \tag{C.6}$$

Por tanto, podemos afirmar que el vector $\mathbf{d} = \hat{\mathbf{s}} - \mathbf{g}$ satisface las condiciones (C.4). Sabemos que $\mathbf{d} \neq 0$ porque $\mathbf{g} \notin K$. Se sigue de (C.5) que

$$\mathbf{d}^T\mathbf{g} = \mathbf{d}^T(\hat{\mathbf{s}} - \mathbf{d}) = (\hat{\mathbf{s}} - \mathbf{g})^T\hat{\mathbf{s}} - \mathbf{d}^T\mathbf{d} = -\|\mathbf{d}\|_2^2 < 0$$

por lo que $\mathbf{d}$ satisface la primera propiedad de (C.4).

De (C.6) tenemos que $\mathbf{d}^T\mathbf{s} \geq 0$ para todo $\mathbf{s} \in K$, por lo que

$$\mathbf{d}^T\left(B\mathbf{y} + C\mathbf{w}\right) \geq 0 \text{ para todo } \mathbf{y} \geq 0 \text{ y para todo } \mathbf{w}$$

Si fijamos $\mathbf{y} = 0$, tenemos que $\left(C^T\mathbf{d}\right)^T\mathbf{w} \geq 0$ para todo $\mathbf{w}$, que se cumple solo si $C^T\mathbf{d} = 0$. Si fijamos $\mathbf{w} = 0$, tenemos que $\left(B^T\mathbf{d}\right)^T\mathbf{y} \geq 0$ para todo $\mathbf{y} \geq 0$, que se cumple solo si $B^T\mathbf{d} \geq 0$. Por tanto, $\mathbf{d}$ también satisface la segunda y tercera propiedad de (C.4) y nuestra demostración queda completa.

■

Aplicando C.3.1 al cono N definido por

$$N = \left\{ \sum_{i \in A(\mathbf{x}^*)} \lambda_i \nabla c_i\left(\mathbf{x}^*\right), \quad \lambda_i \geq 0 \text{ para } i \in \mathcal{A}\left(\mathbf{x}^*\right) \cap \mathcal{I} \right\}$$

y fijando $g = \nabla f(\mathbf{x}^*)$, tenemos que, o bien

$$\nabla f\left(\mathbf{x}^*\right) = \sum_{i \in \mathcal{A}(\mathbf{x}^*)} \lambda_i \nabla c_i\left(\mathbf{x}^*\right) = A\left(\mathbf{x}^*\right)^T\lambda^*, \quad \lambda_i \geq 0 \text{ para } i \in \mathcal{A}\left(\mathbf{x}^*\right) \cap \mathcal{I} \tag{C.7}$$

o bien existe una dirección d tal que $d^T\nabla f(\mathbf{x}^*) < 0$ y $d \in \mathcal{F}(\mathbf{x}^*)$.

D

Códigos utilizados en el desarrollo del trabajo

Código D.1: Código de Python para la resolución de los ejemplos del capítulo 3.

```
# Autores: Laura Medina Henche y Emanuele Schiavi.
# Universidad Rey Juan Carlos, 2024.
# El presente software se distribuye segun la licencia
# GPLv3 (https://www.gnu.org/licenses/gpl-3.0.en.html)


from sympy import *
from sympy.plotting import plot3d, PlotGrid

a, t, T, x0, i, k = symbols('a t T x0 i k')
b = Symbol('b', nonzero=True, noninfinite=True)

# Definimos los parametros
p = [a, b]

# Declaramos las funciones que utilizaremos para la EDO
x1 = Function('x1')
dx1 = x1(t).diff(t)

x2 = Function('x2')
dx2 = x2(t).diff(t)

x = Matrix([x1(t), x2(t)])
dx = Matrix(x.diff(t))
dxp = Matrix([diff(x, v) for v in p])
```

```
26
27 y1 = sin(t)
28 dy1 = diff(y1, t)
29
30 y2 = cos(t)
31 dy2 = diff(y2, t)
32
33 y = Matrix([y1, y2])
34 dy = diff(y, t)
35 dyp = Matrix([diff(y, v) for v in p])
36
37 # Definimos la funcion lambda para el adjunto
38 lmbd1 = Function('lambda1')
39 lmbd2 = Function('lambda2')
40 lmbd = Matrix([lmbd1(t), lmbd2(t)])
41 lmbdT = Matrix([lmbd[i].subs({t: T}) for i in ...
       range(len(lmbd))])
42
43 # Fijamos la funcion f(x,p,t) a integrar
44 lista = [sympify(Pow((x[i] - y[i]), 2), Function) for i in ...
       range(len(x))]
45 suma = 0
46 for i in range(len(lista)):
47     suma = suma + lista[i]
48 f = (1/2) * suma
49 print("f= ", f)
50
51 # Calculamos las derivadas parciales de f
52 dfx = diff(f, x)
53 print("dfx= ", dfx)
54 dfp = Matrix([diff(f, v) for v in p])
55 print("dfp= ", dfp)
56
57 # Fijamos la restriccion explicita dada por la EDO
58 h_exp = Matrix([b * fun for fun in x])
59 print("h_exp= ", h_exp)
60
61 # Fijamos la EDO implicita
62 h = dx - h_exp
63 print("h= ", h)
64
65 # Calculamos las parciales
66 dhx = []
67 dhdx = []
68 dhp = []
69 for i in range(len(h)):
70     dhx.append([diff(h[i], v) for v in x])
71     dhdx.append([diff(h[i], v) for v in dx])
72     dhp.append([diff(h[i], v) for v in p])
73 dhx = Matrix(dhx)
74 dhdx = Matrix(dhdx)
75 dhp = Matrix(dhp)
76 print("dhx= ", dhx)
77 print("dhdx= ", dhdx)
```

```
78 print("dhp= ", dhp)
79
80 # Fijamos la funcion g de restricciones
81 x0 = Matrix([x[i].subs({t: 0}) for i in range(len(x))])
82 print("x0= ", x0)
83 g = Matrix([x0[i] - a for i in range(len(x0))])
84 print("g= ", g)
85
86 # Calculamos sus parciales
87 dgx0 = []
88 dgp = []
89 for i in range(len(h)):
90     dgx0.append([diff(g[i], v) for v in x0])
91     dgp.append([diff(g[i], v) for v in p])
92 dgx0 = Matrix(dgx0)
93 dgp = Matrix(dgp)
94 print("dgx0= ", dgx0)
95 print("dgp= ", dgp)
96
97 # Ahora que tenemos las primeras variables definidas,
98 # resolvemos la primera EDO, la principal
99 ics = [{x0[i]: a} for i in range(len(x0))]
100 resODE = Matrix([simplify(dsolve(h[i], ics=ics[i])).rhs for ...
       i in range(len(h))])
101 resODE = Matrix([sympify(resODE[i], Function) for i in ...
       range(len(resODE))])
102 print("Resultado primera EDO: ", resODE)
103
104 # Vamos al segundo paso del algoritmo
105 ics = [{lmbdT[i]: 0} for i in range(len(lmbdT))]
106 EDO_adj = dfx.subs({x[n]: resODE[n] for n in ...
       range(len(resODE))}).T + lmbd.T * (dhx - dhdx.diff(t)) - \
107             lmbd.diff(t).T * dhdx
108 print("EDO_adj: ", EDO_adj)
109 res_adj = ...
       Matrix([sympify(factor(simplify(dsolve(EDO_adj[i], ...
       ics=ics[i])).rhs), Function) for i in range(len(resODE))])
110 print("Resultado ecuacion del adjunto: ", res_adj)
111
112
113 # Tercer paso del algoritmo
114 dfp = dfp.T
115 dFpCal = []
116 for k in range(len(p)):
117     f_int = dfp[:, k] + res_adj.T * dhp[:, k]
118     f_int = f_int.subs({x[k]: resODE[k] for k in ...
           range(len(resODE))})
119     integral = integrate(f_int, (t, 0, T), conds='none')
120     Fp = integral + res_adj.T.subs(t, 0) * dhdx.subs(t, 0) ...
           * dgx0**-1 * dgp[:, k]
121     dFpCal.append(factor(simplify(factor(Fp))))
122
123 dFpCal = Matrix(dFpCal)
124 print("DFPCal= ", dFpCal)
```

```
125
126
127 # Calculo directo del funcional
128 Fp = []
129 print("Calculamos la integral directamente: ")
130 Fun = integrate(f.subs({x[k]: resODE[k] for k in ...
       range(len(x))}), (t, 0, T), conds='none')
131 Fun = sympify(simplify(Fun), Function)
132 print("Fp = ", Fun)
133 print("Comprobamos que el resultado es correcto...")
134 for v in p:
135     dif = simplify(diff(Fun, v))
136     Fp.append(dif)
137     print("dFp", v, " directo= ", dif)
138
139 # Creamos nuestros graficos
140 t1 = "Integral directa"
141 plot3d(Fp[i].subs({T: 2}), (p[0], 0.001, 2), (p[1], 0.001, ...
       2), title=t1, xlabel=p[0], ylabel=p[1])
142
143 t2 = "Primer componente del\ngradiente"
144 pa = plot3d(dFpCal[0].subs({T: 2}), (p[0], 0.001, 2), ...
       (p[1], 0.001, 2), title=t2, xlabel=p[0], ylabel=p[1],
145             show=False)
146 t3 = "Segundo componente del\ngradiente"
147 pb = plot3d(dFpCal[1].subs({T: 2}), (p[0], 0.001, 2), ...
       (p[1], 0.001, 2), title=t3, xlabel=p[0], ylabel=p[1],
148             show=False)
149 PlotGrid(1, 2, pa, pb, margin=0.7)
150
151 t4 = "Modulo del gradiente"
152 modulo = sqrt((dFpCal[0].subs({T: 2})) ** 2 + ...
       (dFpCal[1].subs({T: 2})) ** 2)
153 plot3d(modulo, (p[0], 0.001, 2), (p[1], 0.001, 2), ...
       title=t4, xlabel=p[0], ylabel=p[1])
```