Ingeniería Matemática
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS UNIVERSIDAD DE CHILE

Entendiendo Aprendizaje Profundo como un Problema de Control Óptimo

MA4703 Control Óptimo: Teoría y Laboratorio

Reporte final proyecto

Integrantes: Daniel Minaya, Felipe Rivera, Felipe Urrutia
Profesor: Héctor Ramírez C. **Auxiliar:** Javier Madariaga R., Pablo Araya Z.
Fecha: 1 de diciembre de 2022

Motivación

El problema de clasificación es una de las tareas más estudiadas en el área de aprendizaje de máquinas (Bishop, C. M. 2006). Modelos lineales se comportan satisfactoriamente cuando los datos del problema son linealmente separables. Sin embargo, no todos los conjuntos de datos cumplen la separabilidad con el uso de un hiperplano de separación (p.ej. pensar en un anillo con un circulo dentro). Por esto, una serie de otras técnicas han sido desarrolladas para resolver el problema de clasificación sobre un conjunto más amplio de datos.

Las técnicas más clásicas y satisfactorias son aquellas basadas en árboles, como los árboles de decisión, y otras basadas en núcleos, como las máquinas de soporte vectorial. Actualmente, las herramientas clásicas han sido superadas en desempeño en diversas áreas de especialización, con el uso de redes neuronales con aprendizaje profundo. Sin embargo, estos nuevos tipos de modelos son considerados como cajas negras, es decir, modelos ocultos que no muestran intrínsecamente como utilizan la entrada para producir la salida. Adicionalmente, debido a la no convexidad de la función objetivo a minimizar y como los datos suelen estar en alta dimensión, no hay resultados que garanticen estabilidad de los óptimos encontrados para los parámetros de las redes neuronales (Benning, M. 2019).

Debido a que los modelos clásicos, algunos interpretables (p.ej. arboles de decisión) y otros estables (p.ej. maquinas de soporte vectorial), logran un desempeño inferior a los modelos cajas negra basados en aprendizajes profundo sin garantías de estabilidad, se estudia el problema de clasificación como un problema de control óptimo, para encontrar modelos con las mejores propiedades. La idea es utilizar los resultados matemáticos obtenidos en la teoría de control óptimo, para resolver el problema de clasificación binaria con un desempeño competitivo, de forma transparente y con garantías de estabilidad. Con esto, tal como nos sugiere la Figura 1, nos enfocaremos en resolver la siguiente pregunta:

¿Podemos entender aprendizaje profundo como un problema de control óptimo para resolver la tarea de clasificación binaria, y obtener modelos con estabilidad e interpretabilidad como los modelos clásicos con un desempeño similar a los modelos modernos?

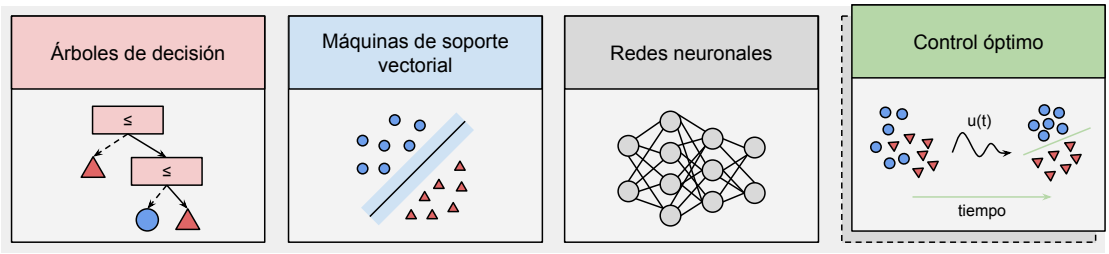

Figura 1: Ilustración de métodos clásicos y modernos utilizados para el problema de clasificación junto a técnicas basadas en control óptimo.

Introducción

Descripción del problema

Queremos resolver un problema de clasificación binaria visto como un problema de control óptimo. Para esto, supondremos que tenemos dos etiquetas c^0 y c^1, y m datos etiquetados $\{(x_i, c_i)\}_{i=1}^m$, donde $c_i \in \{c^0, c^1\}$.

Nuestro objetivo será encontrar parámetros u, W, μ tales que se minimice la siguiente función de costos

$$\frac{1}{2}\sum_{i=1}^{m} |\mathcal{C}(Wh(x_i; u) + \mu) - c_i|^2, \tag{1}$$

donde $h : \mathbb{R}^n \to \mathbb{R}^n$ es una función parametrizada por u, $W \in \mathbb{R}^{1\times n}$ es un vector de pesos, $\mu \in \mathbb{R}$ es un sesgo relacionado al modelo con pesos, y $\mathcal{C} : \mathbb{R} \to \mathbb{R}$ es una función de hipótesis que manda el modelo con peso y sesgo al conjunto discreto de etiquetas $\{c^0, c^1\}$.

Para realizar esto, se hará uso de la arquitectura Residual neural Network (ResNet), donde definiremos la función h como la salida entregada por la ResNet. Con esto, queremos minimizar la siguiente función objetivo

$$\frac{1}{2}\sum_{i=1}^{m} |\mathcal{C}(Wy_i^{[N]} + \mu) - c_i|^2, \tag{2}$$

donde N es el número de capas de la ResNet, sus parámetros son de la forma

$$u = \left(u^{[0]}, \dots, u^{[N-1]}\right),\ u^{[j]} = \left(K^{[j]}, \beta^{[j]}\right) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n \tag{3}$$

y la variable de estado de la ResNet viene dada por

$$y = \left(y^{[0]}, \dots, y^{[N]}\right),\ y^{[j]} = \left(y_1^{[j]}, \dots, y_m^{[j]}\right),\ y_i^{[j]} \in \mathbb{R}^n \tag{4}$$

que satisface

$$y_i^{[j+1]} = y_i^{[j]} + \Delta t f\left(y_i^{[j]}, u^{[j]}\right),\ y_i^{[0]} = x_i \tag{5}$$

Se utilizará $f(y_i^{[j]}, u^{[j]}) = \sigma\left(K^{[j]}y_i^{[j]} + \beta^{[j]}\right)$, con σ una función de activación que actúa coordenada a coordenada.

Si consideramos W, μ fijos, entonces podemos escribir el problema anterior como un problema de control óptimo, definiendo $y_i = y_i(t)$ y $u = u(t) = (K(t), \beta(t))$ como funciones del tiempo para $t \in [0, T]$. Además, notemos que la ecuación (5) es la discretización de Euler en $[0, T]$ de la EDO con paso Δt

$$\frac{d}{dt}y_i(t) = f(y_i(t), u(t)),\ y_i(0) = x_i \tag{6}$$

De esta forma, el problema de control óptimo a resolver es un problema de Mayer y está dado por

$$\begin{aligned} \min_{u(\cdot)}\ & \mathcal{J}(y(T)) = \frac{1}{2}\sum_{i=1}^{m} |\mathcal{C}(Wy_i(T) + \mu) - c_i|^2 \\ \text{s.a. } & \dot{y}_i(t) = f(y_i(t), u(t)) \text{ en } [0, T] \\ & y_i(0) = x_i \end{aligned} \tag{7}$$

Resultados Teóricos

Principio de Pontryagin

Por simplicidad, en esta sección supondremos que $m = 1$, pero el análisis seguirá siendo válido para cualquier m, pues basta sumar la contribución de cada dato. De esta forma, el problema de Mayer viene dado por

$$\begin{aligned} &\min_{u(\cdot)} \ \mathcal{J}(y(T)) \\ &\text{s.a. } \dot{y} = f(y,u) \text{ en } [0,T] \\ &\qquad y(0) = x \end{aligned} \tag{8}$$

donde $\mathcal{J}(z) = \frac{1}{2}|\mathcal{C}(Wz+\mu) - c|^2$ es de clase C^1.

El Hamiltoniano asociado a este problema de Mayer viene dado por

$$H(y,p,u) = p^T f(y,u) = p^T \sigma(Ky+\beta) \tag{9}$$

donde $u = (K,\beta) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n$. Luego, por el principio de Pontryagin tenemos que si (y,u) es solución óptima de (8), entonces existe $p : [0,T] \to \mathbb{R}^n$ solución del sistema

$$\dot{y} = \partial_p H = f(y,u), \quad \dot{p} = -\partial_y H = -p^T \frac{\partial f}{\partial y}(y,u), \quad 0 = \partial_u H = p^T \frac{\partial f}{\partial u}(y,u) \tag{10}$$

Más aún, como los tiempos inicial y final, y la condición inicial están fijos, entonces tenemos que

$$dg = \partial_y g(y_f) dy_f, \qquad [\Theta(t)dt - p(t)^T dy_t]_{t=0}^{t=t_f} = -p(t_f)^T dy_f \tag{11}$$

Con lo cual se obtiene la siguiente condición de transversalidad

$$p(T) = \frac{\partial \mathcal{J}}{\partial y}(y(T)) \tag{12}$$

Ecuación de Hamilton-Jacobi-Bellman

Consideremos el siguiente problema parametrizado por $t_0 \geq 0,\ y_0 \in \mathbb{R}^n$

$$\begin{aligned} (P_{t_0,y_0}) \quad &\min_{u(\cdot)} \ J_B = \int_{t_0}^{T} e^{-\lambda t} \ell(y(t),u(t))dt + \mathcal{J}(y(T)) \\ &\text{s.a. } \dot{y} = f(y,u) \text{ en } [t_0,T] \\ &\qquad y(t_0) = y_0 \end{aligned} \tag{13}$$

y definamos $V(t_0,y_0)$ la función valor de (P_{t_0,y_0}).

En nuestro caso tenemos que $\ell \equiv 0$ y $\lambda = 0$, por lo que la ecuación de Hamilton-Jacobi-Bellman viene dada por

$$\begin{cases} \partial_t V(t,y_0) + \inf_{w=(K,\beta)}\{\nabla_y V(t,y_0))^T \sigma(Ky_0+\beta)\},\ (t,y_0) \in [0,T)\times\mathbb{R}^n \\ V(T,y_0) = \mathcal{J}(y_0),\ y_0 \in \mathbb{R}^n \end{cases} \tag{14}$$

Resultados numéricos

La preparación de los datos, las implementaciones de los modelos propuestos, el código para el entrenamiento de los modelos, el guardado y las predicciones de los modelos en los conjuntos y el despliegue de los resultados de esta sección están disponibles libremente en el siguiente repositorio de GitHub: https://github.com/furrutiav/deep-learning-as-optimal-control.

Modelos propuestos

Discretizar la dinámica del sistema puede realizarse de diferentes maneras. Por esto, se proponen dos modelos que aproximan la dinámica del sistema con el método de Euler. Estos métodos son **ResNet**, metodo detallado al inicio de este reporte, y **ODENet**, una variante dada por

$$y_i^{[j+1]} = y_i^{[j]} + \alpha_t f\left(y_i^{[j]}, u^{[j]}\right), \; y_i^{[0]} = x_i \tag{15}$$

donde α_t es un parámetro aprendible que reemplaza el parámetro Δt del modelo **ResNet**.

Evaluación

Se realiza una evaluación de los modelos propuesto a partir de diferentes **Conjuntos de estudio**, **Modelos de referencia** para comparar, **Métricas** apropiadas de evaluación y una inspección por **Visualización**.

Conjuntos de estudio. Se proponen cuatros conjuntos de datos obtenidos sintéticamente para el problema de clasificación binaria balanceada, ver Figura 2.

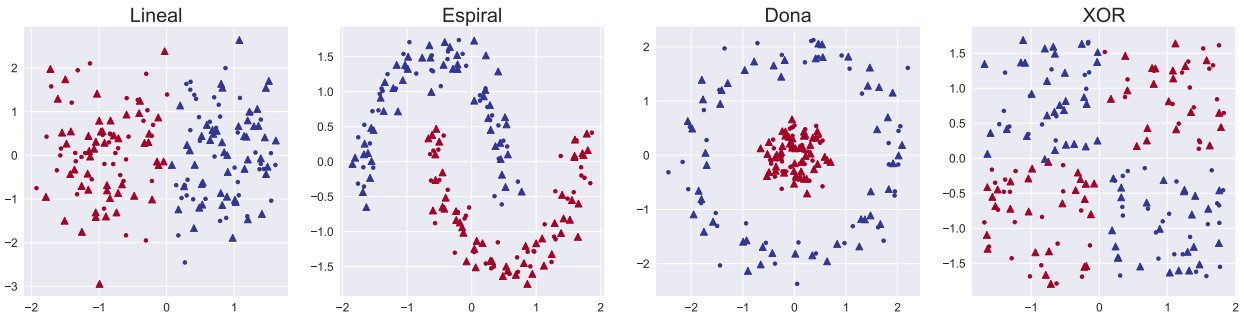

Figura 2: Conjuntos de datos de estudio, de izquierda a derecha Lineal, Espiral, Dona y XOR. (**Rojo**) Clase positiva y (**Azul**) clase negativa. (**Triangulos**) Datos de entrenamiento y (**Puntos**) datos de prueba.

Los cuatro conjuntos de datos seleccionados permiten evaluar la capacidad de los modelos propuestos para encontrar una región de decisión desde simple hasta compleja. El conjunto de datos *Lineal* es el mas simple. Mientras que los conjuntos de datos *Espiral*, *Dona* y *XOR* aumentan en dificultad, dado que no basta con un modelo lineal para separar las clases de clasificación.

Cada conjunto de datos posee un subconjunto de entrenamiento con 100 datos y otro subconjunto de prueba con 80 datos, con intersección vacía. El propósito de cada conjunto es utilizarlos de forma separada, uno para ajustar un modelo de clasificación binaria y otro para medir su desempeño sobre un conjunto distinto al de entrenamiento.

Modelos de referencia. Para comparar los modelos propuestos se utilizarán cuatros modelos clásicos y modernos utilizados para resolver el problema de clasificación binaria. Estos son:

- *DT*. Árbol de decisión, el modelo más simple basado en arboles con decisiones If/Else. Se usa profundidad menor a 10.
- *SVC*. Maquina de soporte vectorial para clasificación, basado en kernel con el objetivo maximizar el margen de decisión. Se usa un kernel Gaussiano, RBF.
- *MLP*. Perceptron multi-capa, red neuronal con solo una capa oculta de tamaño 100.

- *NN*. Red neuronal de avance totalmente conectada, donde cada capa oculta tiene tamaño 2 y profundidad 5.

Métricas. Como el problema es balanceado, se utiliza el accuracy como métrica de desempeño para comparar los modelos. Accuracy corresponde a la proporción entre la cantidad de datos bien predichos y la cantidad total de datos evaluados, es decir,

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN}.$$

Adicionalmente se cuantifica el tiempo de entrenamiento de cada modelo sobre cada conjunto de estudio y el tiempo de inferencia (o predicción) sobre los subconjuntos de prueba. De esta manera, comparamos la complejidad de los modelos propuestos contra los modelos de referencia.

Visualización. Se grafican las regiones de decisión encontradas por cada modelo propuesto y de referencia. Además, para los modelos propuestos, basados en control óptimo, se dibujan las trayectorias que siguen los datos para luego ser separados linealmente.

Desempeño y complejidad

El desempeño de cada modelo sobre cada conjunto de estudio, tanto en entrenamiento como prueba, se presenta en la Tabla 1. Para cada conjunto de estudio, se marcan en negrita aquellos valores que obtienen el valor máximo en el subconjunto de prueba.

Tabla 1: Métricas de accuracy para todos los modelos y conjuntos de datos tanto en el conjunto de entrenamiento (train) como de prueba (test).

Dataset	**Lineal**		**Espiral**		**Dona**		**XOR**	
Set	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
DT	100.0	**100.0**	100.0	97.50	100.0	90.00	98.0	95.00
SVC	99.0	98.75	100.0	**100.0**	100.0	**100.0**	100.0	**98.75**
MLP	100.0	**100.0**	89.0	87.50	99.0	98.75	99.0	**98.75**
NN	99.0	98.75	86.0	86.25	69.0	68.75	56.0	48.75
ResNet	100.0	**100.0**	100.0	98.75	100.0	98.75	96.0	88.75
ODENet	100.0	**100.0**	100.0	98.75	100.0	**100.0**	100.0	96.25

El tiempo de entrenamiento de cada modelos sobre cada conjunto de estudio, se presenta en la Tabla 2.

Tabla 2: Tiempo de entrenamiento en segundos, para todos los modelo y cada conjunto de estudio de entrenamiento.

Dataset	**Lineal**	**Espiral**	**Dona**	**XOR**
DT	0.001	0.001	0.000	0.001
SVC	0.000	0.001	0.000	0.001
MLP	0.088	0.092	0.087	0.091
NN	0.136	0.137	0.139	0.139
ResNet	33.598	57.725	184.400	430.618
ODENet	6.501	29.140	28.635	62.311

El tiempo de inferencia (o predicción) de cada modelos sobre cada conjunto de estudio, se presenta en la Tabla 3.

Tabla 3: Tiempo de inferencia (predicción) en milésimas de segundos, para todos los modelo y cada conjunto de estudio de entrenamiento.

Dataset	**Lineal**	**Espiral**	**Dona**	**XOR**
DT	0.000	0.000	0.000	0.000
SVC	0.000	1.000	0.000	0.000
MLP	0.000	0.000	0.000	0.000
NN	0.000	0.000	0.000	0.000
ResNet	10.010	10.009	11.010	11.010
ODENet	5.004	4.004	4.003	4.003

Región de decisión y trayectorias

Las regiones de decisión obtenidas por cada modelo sobre cada conjunto de estudio junto a los datos de prueba se presentan en la Figura 3. Regiones en rojo correspondes aquellas zonas donde el modelo predice con la clase positiva, mientras que en azul con la clase negativa.

Para los modelos propuestos, basados en control óptimo, ResNet y ODENet se dibujan las trayectorias de los datos, ver Figura 4.

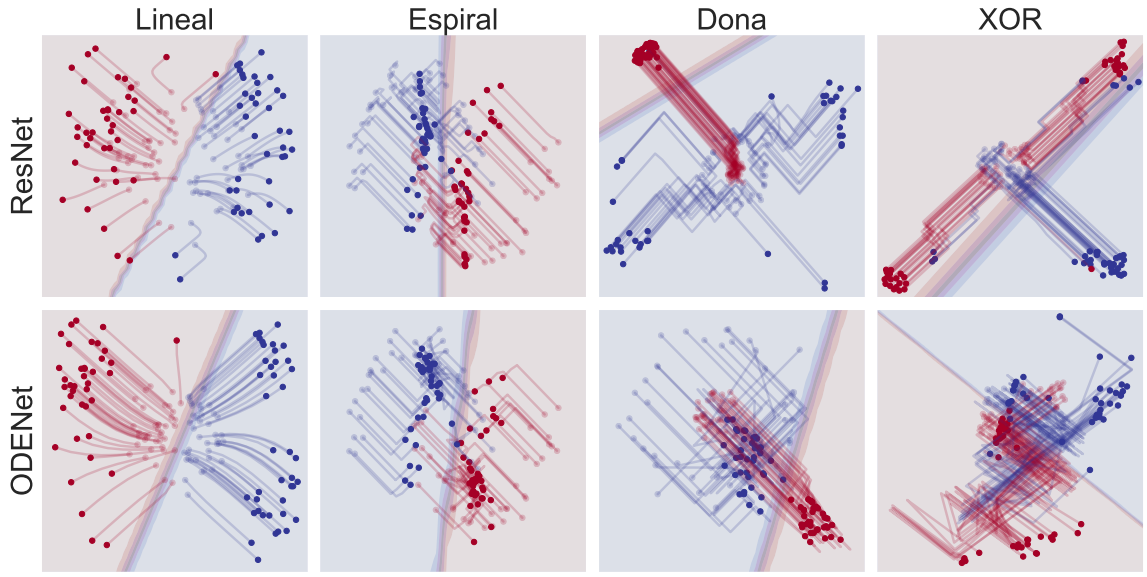

Figura 4: Trayectorias para cada punto de los datos de prueba y plano de decisión de los modelos ResNet y ODENet, para cada conjunto de estudio, ver Figura 2 para datos y Figura 3 para modelos. (**Columna**) Datos de estudio y (**Fila**) Modelos ResNet y ODENet.

Discusión de resultados

Resultados Teóricos

Principio de Pontryagin. Con este principio se obtiene un sistema de ecuaciones diferenciales ordinarias dadas por la ecuación (10), el cual puede discretizarse para obtener el control óptimo discretizado como se sugiere en Benning, M. (2019). Otra alternativa, es resolverlo utlizando el método de tiro, ya que se tiene una condición inicial $y(0)$ sobre y y una condición final $p(T)$ sobre p, de modo que el sistema adjunto obtenido llegue a $p(T)$ en el tiempo T, y luego utilizar el método `solve_ivp` de la librería `scipy` para obtener las

soluciones para $y(\cdot)$ y $p(\cdot)$.

Ecuación de Hamilton-Jacobi-Bellman. Como en este caso se tiene una ecuación en derivadas parciales, se debe discretizar tanto en tiempo como en espacio, para luego utilizar herramientas de análisis numérico de EDPs y resolver la ecuación.

Resultados numéricos

Desempeño. De acuerdo a la Tabla 1, para el conjunto de datos *Lineal* en el subconjunto de prueba, todos los modelos, tanto propuestos como de referencia, obtienen un desempeño máximo excepto SVC (maquina de soporte vectorial) y NN (red neuronal).

Sin embargo, en el conjunto de datos *Esprial*, el modelo de referencia SVC es el único que obtiene un desempeño máximo. Pese a eso, en el mismo conjunto de datos, los modelos basados en control óptimo, ResNet y ODENet solo se equivocan en uno de los datos de prueba obteniendo un 98.75 % de accuracy.

Por otra lado, el conjunto de datos *Dona* es resuelto satisfactoriamente tanto por ODENet como SVC, es decir, un 100 % de accuracy. Mientras que el modelo ResNet se equivoca en solo uno de los datos de prueba.

Ahora bien, para el conjunto de datos *XOR*, el desempeño máximo es de 98.75 %, es decir, todos los datos de prueba bien predichos excepto uno, con los modelos SVC y MLP (Perceptron multi-capa). Mientras que, para este conjunto de datos, el mejor modelo basado en control óptimo es ODENet, con un 96.25 % de accuracy.

Globalmente se observa que, para los modelos basados en control óptimo, el modelo ODENet es mejor que el modelo ResNet, pero no significativamente. Por otro lado, en comparación con los modelos de referencia, el modelo ODENet supera a todo ellos de forma significativa exceptuando al modelo SVC.

Complejidad. De acuerdo a la Tabla 2, el tiempo de entrenamiento de los modelos de referencia es considerablemente menor que el tiempo requerido para encontrar los controles óptimos para los modelos propuestos (ResNet y ODENet).

En particular, los modelos de referencia no superan los 0.1 segundos para ser entrenados. Mientras que los modelos basados en control óptimo, algunos llegan a demorarse más de un minuto. Sin embargo, este tiempo es considerablemente menor para el modelo ODENet cuando se compara con el tiempo utilizado por el modelo ResNet. Esto ultimo se debe a que modelo ODENet posee menos parámetros que el modelo ResNet, donde el modelo ResNet es una discretización de Euler con 15 pasos y ODENet con solo 5.

Adicionalmente, que los modelos de referencia no superen 1 segundo para ser entrenados, se debe principalmente a que los modelos de referencia son implementados utilizando la librería `sklearn`, diseñada específicamente para ajustar de la manera más eficiente dichos modelos. Mientras que los modelos propuestos son implementados por nosotros con el método `minimize` de la librería `scipy`, con potenciales debilidades de diseño, es decir, implementación que puede ser afinada y que puede mejorar los resultados obtenidos tanto en tiempo como en desempeño.

De acuerdo a la Tabla 3, el tiempo de inferencia (o predicción) de los modelos basados en control óptimo es mayor que el tiempo gastado por los modelos de referencia. Sin embargo, esta diferencia es despreciable, considerando que las mediciones están en el orden de mili-segundos.

Regiones de decisión. De acuerdo a la Figura 3, para el conjunto de datos *Lineal* todos lo modelos, tanto los de referencia como los propuestos, logran encontrar una región de decisión simple y apropiada. No se observa mayor diferencia entre ellas.

Por otro lado, para el conjunto de datos *Espiral*, las regiones de decisión cambian entre los modelos. Los modelos MLP y NN, no logran encontrar una región de decisión apropiada, lo que concuerda con el desempeño obtenido por estos modelos en el subconjunto de prueba. El modelo DT (árbol de decisión) encuentra un margen con esquinas rectangulares, mientras que el modelo SVC logra un margen más suave. Incluso los modelos basados en control óptimo, ResNet y ODENet, logran encontrar un margen satisfactorio similares entre si pero un poco más estrecho que el encontrado por el modelo SVC.

Ahora bien, para el conjunto de datos *Dona*, existe una mayor diferencia entre las regiones de decisión, debido a que estos datos de estudio son más difíciles que los primeros dos estudiados, *Lineal* y *Espiral*. El modelo DT, no encuentra un región natural, el margen es un rectángulo mientras que los datos al interior del anillo son un circulo. Sin embargo, el modelo SVC, logra encontrar un margen de decisión más apropiado y suave. Así mismo, el modelo MLP, pero con una región no simétrica. Por ultimo, el modelo NN no encuentran

una región satisfactoria. En cambio, los modelos propuestos ResNet y ODENet, logran encontrar un región decisión suficiente para resolver el problema de clasificación. Pero, solo el modelo ODENet genera una región de decisión natural circular y no así el modelo ResNet con una mancha que nace en el centro y sale por uno de los extremos del anillo.

Finalmente, para el conjunto de datos *XOR*, se obtienen regiones más complejas y diferentes entre los modelos de referencia y propuestos. Primero, solo los modelos DT, SVC y MLP, generan una región de decisión apropiada, pero no así el modelo NN. Por otro lado, los modelo ResNet y ODENet, encuentran regiones de decisión muy diferentes, donde el margen de decisión del modelo ResNet es más suave que el margen del modelo ODENet. Sin embargo, ambos modelos encuentran regiones de decisión suficientes.

Trayectorias. De acuerdo a la Figura 4, para el conjunto de datos *Lineal*, tanto el modelo ResNet como ODENet producen trayectorias similares. Ocurre que los controles óptimos encontrados por ambos modelos producen que los datos de prueba se muevan desde adentro hacia afuera. De esta forma, es más fácil separar linealmente los datos con un hiperplano de separación.

Para el conjunto de datos *Espiral*, ocurre de forma similar que ambos modelos encuentran controles óptimos que producen un comportamiento parecido en las trayectorias. Sin embargo, este comportamiento es diferente al observado en el conjunto de datos *Lineal*. Esta vez, los datos de prueba que están en los brazos del espiral pero afuera del centro son atraídos hacia el centro. En cambio, los datos de prueba que están cerca del bulbo del espiral son repelidos desde el centro hacia afuera. De esta forma, los datos se mueven a posiciones donde esta vez los datos son linealmente separables.

Ahora bien, para el conjunto de datos *Dona*, los controles óptimos de los modelos ResNet y ODENet producen trayectorias diferentes sobre los datos. Primero, el control óptimo del modelo ResNet, mueve los datos que están en el centro hacia afuera arriba a la izquierda sin desagruparlos. En cambio, los datos que están en el anillo son repelidos desde afuera hacia adentro prefiriendo mover arriba a la derecha y abajo a la izquierda. De este modo, las nuevas posiciones de los datos permiten que estos sean separados linealmente. Por otro lado, el control óptimo del modelo ODENet, mueve de manera distinta los datos. Los datos que están en el centro se mueven hace abajo a la derecha sin desagruparlos. Mientras que los datos que están en el anillo se agrupan en la misma dirección y sentido que los datos del centro pero de forma más lenta. Estas nuevas posiciones, también permiten que los datos sean separados linealmente.

Finalmente, para el conjunto de datos *XOR*, los controles óptimos encontrados por los modelos ResNet y ODENet generan trayectorias significativamente distintas. Primero, el control óptimo del modelo ResNet, mueve los datos del primer cuadrante hacia arriba a la derecha y los del tercer cuadrante hacia abajo a la izquierda. Mientras que los datos ubicados en el segundo y cuarto cuadrante se dirigen hacia abajo a la derecha. Esto permite que las trayectorias entre datos con diferentes clases de clasificación no se intersecten y que además las posiciones finales permitan separar linealmente los datos. Por otro lado, el control óptimo del modelo ODENet, dirige simétricamente los datos en azul y rojo con respecto a la diagonal que va del segundo cuadrante al cuarto cuadrante. Sin embargo, el comportamiento de las trayectorias no es claro.

Conclusiones

En este trabajo se estudia el aprendizaje profundo como un problema de control óptimo para resolver el problema de clasificación binaria. En relación a otras técnicas y métodos para resolver el problema de clasificación, se plantea el interés de utilizar control óptimo para obtener modelos con un buen desempeño, capaces de ser fácilmente interpretados y que garanticen estabilidad para encontrar una solución óptima. Para estudiar esto, se obtienen resultados teóricos con el principio de Pontryagin y la ecuación de Hamilton-Jacobi-Bellman, junto a resultados numéricos con el siguiente esquema de evaluación:

Se proponen dos métodos basado en control óptimo, ResNet y ODENet, ambos métodos son discretización tipo Euler de la dinámica asociada al problema de control óptimo. Se compara el comportamiento de estos métodos contra otros de referencias, con aquellos que llamamos clásicos, arboles de decisión y maquinas de soporte vectorial, y otros que llamamos moderno, redes neuronales. Para evaluar esto, se dispone de cuatros conjuntos de datos generados sintéticamente para medir la capacidad predicativa de los métodos propuestos contra los de referencia, con datos con dificultad desde fácil hasta difícil. Para medir el desempeño, se evalúan los métodos con la métrica de accuracy sobre un subconjunto de prueba. Adicionalmente, se estima el tiempo

utilizado por los modelos durante el tiempo de entrenamiento e inferencia de forma separada. Finalmente, se realiza un estudio visual de las regiones de decisión encontradas por todos los métodos y las trayectorias generadas por los métodos basados en control óptimo.

Primero, de los resultados teóricos se obtiene que, el principio de Pontryagin y la ecuación de Hamilton-Jacobi-Bellman, cada uno nos permiten asegurar la existencia de un control óptimo bajo ciertas condiciones. De esta forma, con los algoritmos ResNet y ODENet obtendremos un resultado óptimo independiente del valor de T fijado. Además, nos entregan una expresión para calcular el control óptimo en función de los parámetros del problema de minimización, lo cual nos permite darle una mayor interpretación al control obtenido y entender de mejor manera la naturaleza de los datos.

Luego, de los resultados numéricos podemos ver que se logra el objetivo inicial planteado. Los métodos ResNet y ODENet brindan una mayor interpretabilidad a través de las trayectorias obtenidas, evidenciando el comportamiento interno de los modelos basados en control óptimo. Además, logran un desempeño competitivo con respecto a los modelos de referencia utilizados, llegando incluso a ser mejor en algunos casos.

Sin embargo, este enfoque mediante control óptimo posee ciertas limitaciones, siendo la más notoria su complejidad. Tanto el tiempo de entrenamiento como el tiempo de inferencia son muy altos en comparación al resto de métodos de clasificación utilizados, aunque se deben hacer ciertas observaciones en este punto. Primero, las implementaciones de ResNet y ODENet pueden ser optimizadas para mejorar la complejidad. Segundo, los otros métodos de clasificación utilizados son implementaciones ya optimizadas pertenecientes a la librería `sklearn`, por lo cual esta comparación puede llegar a ser injusta.

Pese a las limitaciones encontradas, consideramos que los resultados son interesantes y estimulantes. Logramos encontrar métodos con una capacidad predicativa similar a los modelos basados en aprendizaje profundo pero con la posibilidad de estudiar internamente como utilizan los datos para realizar las predicciones a partir de trayectorias. Los resultados teóricos y numéricos nos sugieren estudiar, en un trabajo futuro, el problema de control óptimo con tiempo final libre, permitiendo una profundidad variable, y un costo integral sobre el control para penalizar la energía de la dinámica, permitiendo trayectorias menos intrincadas. Adicionalmente, estos resultados preliminares con dinámicas temporales, nos siguieren estudiar el caso de considerar derivada en espacio, con ecuaciones en derivada parcial y su equivalencia con los modelos de redes neuronales de convolución (Alt, T. 2022)

Referencias

1. Bishop, C. M., & Nasrabadi, N. M. (**2006**). Pattern recognition and machine learning. *New York: springer*, 4(4), 738.

2. Benning, M., Celledoni, E., Ehrhardt, M. J., Owren, B., & Schönlieb, C. B. (**2019**). Deep learning as optimal control problems: Models and numerical methods. *arXiv preprint* https://arxiv.org/abs/1904.05657.

3. Alt, T., Schrader, K., Augustin, M., Peter, P., & Weickert, J. (**2022**). Connections between numerical algorithms for PDEs and neural networks. *Journal of Mathematical Imaging and Vision*, 1-24.

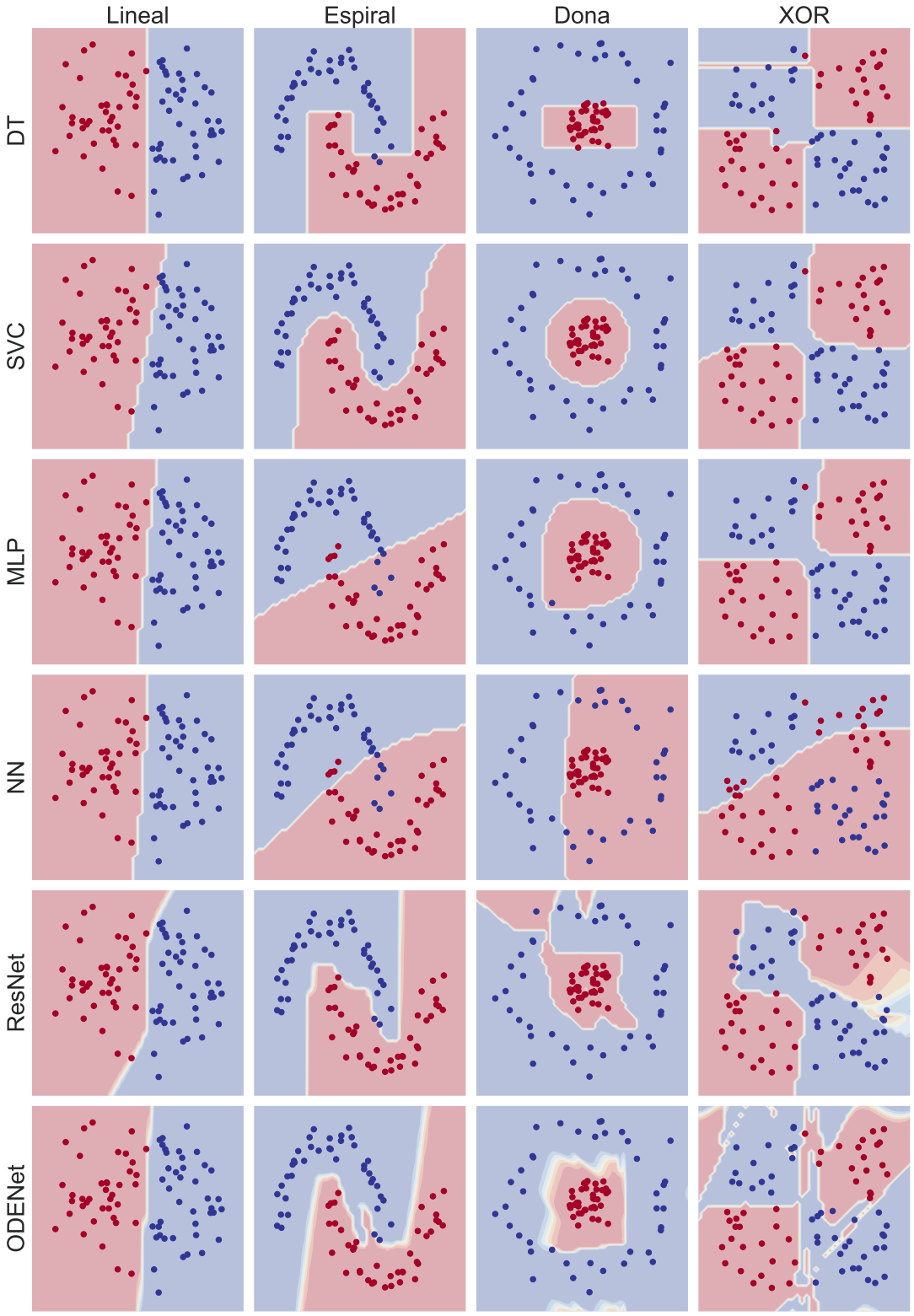

Figura 3: Región de decisión de los modelos entrenados y puntos con los datos de prueba, para cada conjunto de estudio, ver Figura 2 para datos. (**Columna**) Datos de estudio y (**Fila**) Modelos de referencia y propuestos.