Entendiendo Deep Learning como un problema de Control Óptimo

Fabián Ulloa

8 Diciembre 2023

1. Introducción

La clasificación binaria, uno de los problemas más importantes en el campo de machine learning, ha sido objeto de extenso estudio debido a su relevancia en diversas aplicaciones. Si bien los modelos lineales han demostrado su eficacia en casos donde los datos son linealmente separables, la realidad de conjuntos de datos con patrones más intrincados requiere enfoques más sofisticados. Esta diversidad de formas y relaciones en los datos ha impulsado el desarrollo de técnicas más avanzadas y flexibles.

En este contexto, se torna fundamental explorar cómo diferentes algoritmos de clasificación se enfrentan a la diversidad de conjuntos de datos. Modelos clásicos, como el Árbol de Decisión, Support Vector Machine (SVM) y las Redes Neuronales, han sido pilares en la resolución de problemas de clasificación. Sin embargo, cada uno de ellos presenta ventajas y desafíos específicos.

Por ejemplo, los Árboles de Decisión son conocidos por su interpretabilidad y capacidad para manejar relaciones no lineales, pero tienden a sobreajustar datos complejos. Las SVM son efectivas en espacios de alta dimensión y en casos de datos no lineales, pero pueden ser sensibles a la elección del kernel. Las Redes Neuronales, por su parte, son estructuras inspiradas en la organización del cerebro humano. Estas consisten en capas de nodos interconectados, conocidos como neuronas, que procesan la información de entrada para producir una salida. La capacidad distintiva de las redes neuronales radica en su habilidad para aprender automáticamente a partir de los datos, ajustando sus parámetros internos a medida que se exponen a nuevas observaciones. Las Redes Neuronales han emergido como herramientas poderosas para aprender representaciones complejas, pero suelen requerir grandes cantidades de datos y tiempo de entrenamiento.

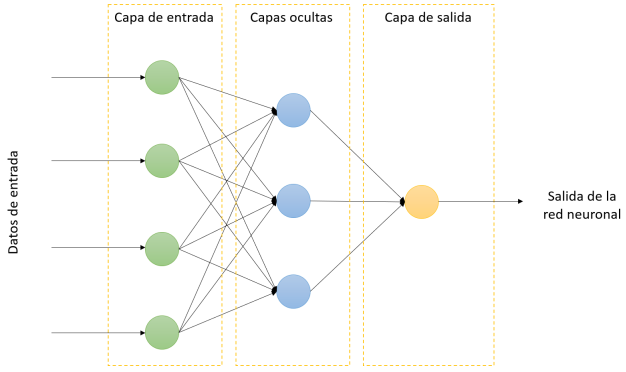

Figura 1: Ilustración de una Red Neuronal

En este contexto, la integración entre el deep learning y el control óptimo ofrece una perspectiva innovadora para abordar la clasificación binaria. El objetivo del presente informe es, por ende, abordar el problema de clasificación binaria desde el ámbito del aprendizaje profundo entendiendolo como un problema de control óptimo. La tarea consiste en desarrollar un clasificador eficaz a partir de un conjunto de datos etiquetados, utilizando un enfoque que integra una Red Neuronal Residual (ResNet) como modelo. Esta ResNet se ajusta a través de la resolución de un problema de control óptimo, donde los parámetros del modelo son optimizados para minimizar una función de pérdida.

2. Descripción del Problema

Problema de Clasificación

Consideremos el problema de encontrar un clasificador a partir de muestras conocidas $\{(x_i, c_i)\}_{i=1}^m$, donde $c_i \in \{c^0, c^1\}$ es la etiqueta de $x_i \in \mathbb{R}^n$. Este problema usualmente se formula como un problema de regresión generalizado:

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

Con $W \in \mathbb{R}^{1\times n}$ vector de pesos,$\mu \in \mathbb{R}$ escalar relacionado al sesgo, $C : \mathbb{R} \to \mathbb{R}$ función de hipótesis y h es una función parametrizada por u que devuelve vectores en $\mathbb{R}^n$. La idea es ocupar h como el output entregado por ResNet. Consideramos entonces el problema

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

donde, para $j = 0, ..., N-1$, los parámetros son de la forma $u^{[j]} := (K^{[j]}, \beta^{[j]}),\ u = (u^{[0]}, \dots, u^{[N-1]})$ con $K^{[j]}$ es una matriz de pesos de dimensión $n \times n,\ \beta^{[j]}$ representa los sesgos, y N es el número de capas. La variable de estado de ResNet viene dada por

$$y = (y^{[0]}, \dots, y^{[N]}),\ y^{[j]} = (y_1^{[j]}, \dots, y_m^{[j]}) \tag{3}$$

tales que

$$y_i^{[j+1]} = y_i^{[j]} + \Delta t \cdot f(y_i^{[j]}, u^{[j]}),\ y_i^{[0]} = x_i. \tag{4}$$

donde se suele usar $f(y_i^{[j]}, u^{[j]}) = \sigma(K^{[j]}x + \beta^{[j]})$ con σ función de activación que actúa componente a componente, $K^{[j]} \in \mathbb{R}^{n\times n}$ y $\beta^{[j]} \in \mathbb{R}^n$. En lo que sigue llamaremos al modelo recientemente descrito como ResNet.

Problema de Control Óptimo

Notemos que el problema anterior es la discretización en el tiempo de un problema de control óptimo cuando los valores de W y μ son fijos. En efecto, escribiendo $y_i = y_i(t)$, luego $y_i^{[j]} = y_i(t_j)$ y $u = u(t) = (K(t), \beta(t)),\ t \in [0, T]$, y con $\{t_j\}_{j=0}^N$ una discretización de $[0, T]$. Entonces, en el continuo, obtenemos el problema de Mayer:

$$\min_u\ J(y(T)) = \sum_{i=1}^{m} |C(Wy_i(T) + \mu) - c_i|^2 \tag{5}$$

$$\text{s.a. } \dot{y_i}(t) = f(y_i(t), u(t)), t \in [0, T],\quad y_i(0) = x_i \tag{6}$$

Pues basta notar que (4) es la discretización de Euler de la EDO en (6).

Consideremos, sin perdida de generalidad, en lo que sigue $m = 1$ (si no, el mismo análisis se puede hacer pero para cada indice i). Notemos que esencialmente tenemos el siguiente problema de control óptimo:

$$\min_u\ J(y(T)) \tag{7}$$

$$\text{s.a. } \dot{y}(t) = f(y(t), u(t)), t \in [0, T],\quad y(0) = x \tag{8}$$

Con esto es claro que el Hamiltoniano del problema viene dado por

$$H(t, y, p_0, p, u) = H(y, p, u) = p^T f(y, u) \tag{9}$$

Por lo que, por el Principio del Máximo de Pontryagin, tendriamos que si $(y(\cdot), u(\cdot))$ es solución del problema, entonces existe $p(\cdot)$ absolutamente continua tal que:

$$\dot{p}(t) = -\partial_y H = -\partial_y f(y(t), u(t))^T p(t) \tag{10}$$

y, como el control es libre, entonces:

$$\partial_u f(y(t), u(t))^T p(t) = 0 \tag{11}$$

Estudiemos ahora las condiciones de Transversalidad, siguiendo la notación usual salvo por la notación de la trayectoria (la cual denotamos por y),tenemos t_0, t_f, y_0 fijos y $g(e[y]) = J(y_f)$, luego

$$dg = \partial_y J(y_f) dy_f, \quad (\Theta(t)dt - p(t)^T dy_t)\Big|_{t_0}^{t_f} = -p(t_f)dy_f$$

Lo que implica

$$p_0 \partial_y J(y_f) = p(t_f)$$

Con $p_0 \geq 0$. Notar que si $p_0 = 0$, lo anterior en conjunto con (9) implica que $p \equiv 0$, una contradicción, por lo que $p_0 \neq 0$, sin pérdida de generalidad $p_0 = 1$, por lo que tenemos:

$$p(T) = \partial_y J(y(T)) \tag{12}$$

Finalmente, obtengamos las ecuaciones de HJB, para ello notamos que un problema de tiempo final fijo, con $\ell \equiv 0, \lambda = 0$ (según la notación usual), luego si $V(x,T)$ es la función valor del problema dado por $(7)-(8)$ entonces obtenemos las ecuaciones de HJB:

$$\partial V_t(x,t) + \inf_{u=(K,\beta)} \nabla_x V(x,t)\sigma(Kx+\beta) = 0, \ (t,x) \in [0,T) \times \mathbb{R}^n \tag{13}$$

$$V(x,T) = J(x), \ x \in \mathbb{R}^n \tag{14}$$

donde se usó (9) evaluado en $p = \nabla_x V(x,t), \ y = x$ y la forma usual de f mencionada en el apartado anterior.

3. Método Numérico

El problema entonces es minimizar (2) sobre $u = (u^{[j]})_{j=0}^{N-1}$, con $u^{[j]} = (K^{[j]}, \beta^{[j]})$, para ello consideramos resolver la restricción de la trayectoria $y = (y_1, \ldots, y_m)$ con el método de Euler, es decir, como lo mostrado en (4). Notamos que además se quiere minimizar sobre los W, μ, pero esto también los podemos tomar en el control u, es decir, minimizar sobre los $u = ((K^{[j]}, \beta^{[j]})_{j=0}^{N-1}, W, \mu)$.

Utilizaremos $N = 10,\ \Delta t = 1,\ \sigma(x) = tanh(x),\ C(x) = 1/(1 + e^{-x})$ donde $\sigma(\cdot)$ actúa componente a componente, además, usaremos minimize de scipy con un control inicial u tal que para todo $j = 0, ..., N-1$:

$$K^{[j]} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ \beta^{[j]} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},\ W = \begin{pmatrix} 1 & 1 \end{pmatrix},\ \mu = 0$$

Consideraremos 4 set de datos de tamaño 250 cada uno (ver Figura 2), estos set de datos son tales que a cada $x_i \in \mathbb{R}^2$ (luego $n = 2$) en la muestra se le asocia una etiqueta en $\{c^0 = 0, c^1 = 1\}$. De cada set de datos, se utilizarán 150 muestras para entrenar el modelo (por entrenar nos referimos al proceso de encontrar el u óptimo), y los 100 datos restantes se usarán para testear el modelo.

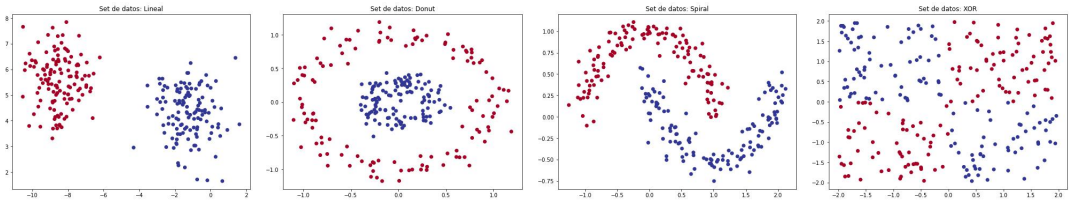

Figura 2: Conjuntos de Set de Datos a utilizar: De izquierda a derecha corresponden a Lineal, Donut, Spiral, XOR. En todos la clase $c^0 = 0$ corresponde a los puntos rojos y $c^1 = 1$ a los azules

Para medir la eficacia el modelo consideramos las siguientes métricas, donde VP = Verdaderos positivos, VN = Verdaderos Negativos, FP = Falsos Positivos y FN = Falsos Negativos:

- Correctitud: $\frac{VP+VN}{VP+VN+FP+FN}$, corresponde al % de acierto del modelo.
- Sensibilidad (Recall): $\frac{VP}{VP+FN}$ corresponde al % de acierto sobre lo realmente éxito.
- Especificidad: $\frac{VN}{VN+FP}$ corresponde al % de acierto sobre lo realmente no éxito.
- Precisión: $\frac{VP}{VP+FP}$ corresponde al % de acierto sobre lo predicho como éxito
- f1-score: $2 \cdot \frac{\text{Precisión} \cdot \text{Recall}}{\text{Precisión} + \text{Recall}}$ es una medida que pondera la sensibilidad con la precisión. Entre más cercano a 1 mejor

Destacar que en este contexto consideramos $c^1 = 1$ = éxito y $c^0 = 0$ = no éxito. Además, compararemos nuestro modelo con otros cuatro modelos conocidos utilizados para resolver el problema de clasificación binaria. Estos son:

- Árbol de Decisión (DCT): Modelo que toma decisiones basadas en características, dividiendo el conjunto de datos en subconjuntos usando una estructura de árbol.
- Análisis del Discriminante Cuadrático (QDA): Método estadístico para clasificación que maximiza la separación entre clases considerando las covarianzas de cada clase.
- Support Vector Machine (SVT): Modelo que encuentra un hiperplano de separación óptimo para maximizar la distancia entre clases.
- Multi-layer Perceptron Classifier (MLP): Red neuronal artificial con capas de nodos interconectados. Para efectos de nuestro proyecto consta de 5 capas de 10 neuronas cada una.

Se utilizaran los mismos datos de entrenamiento y de testeo con estos modelos con tal de obtener las métricas previamente mencionadas para estos modelos. Tambien se medirán los tiempos de ejecución de entrenamiento (para nuestro modelo esto es el tiempo que tarda en minimizar) y el tiempo que se tarda en clasificar (o predecir) los datos de testeo. Finalmente se grafican las regiones de decisión asociadas a cada modelo, y la trayectoria asociada al control óptimo u encontrado en nuestro modelo, esto es, la trayectoria dada por (4) pero con nuestro control óptimo u, este último gráfico contempla solo los datos de entrenamiento.

4. Resultados

A continuación se presentan los indicadores obtenidos según cada modelo y gráficos asociados al modelo ResNet, para cada set de datos considerado. Los gráficos asociados a los demás modelos se encuentran en el anexo. Los códigos relacionados, los cuales contienen la implementación de los modelos mencionados, predicción, testeo, etc., junto con los resultados obtenidos se encuentran en el repositorio de github: https://github.com/fbnljd/Deep-Learning-como-un-problema-de-Control-Optimo

Resultados: Set de Datos Lineal

El control u óptimo encontrado logra que nuestro funcional J alcance un valor de $J = 2{,}63 \times 10^{-7}$.

Por otro lado, la tabla 1 muestra las métricas obtenidas para cada modelo, mientras que la tabla 2 muestra los tiempos de ejecución asociados:

Modelo	**Correctitud**	**Sensibilidad**	**Especificidad**	**Precisión**	**F1-score**
ResNet	99,0 %	98,04 %	100 %	100 %	99,01 %
DCT	100 %	100 %	100 %	100 %	100 %
QDA	100 %	100 %	100 %	100 %	100 %
SVC	100 %	100 %	100 %	100 %	100 %
MLP	99,0 %	98,04 %	100 %	100 %	99,01 %

Tabla 1: Tabla de indicadores de cada Modelo

Tiempo [seg] de	**ResNet**	**DCT**	**QDA**	**SVC**	**MLP**
Entrenamiento	29,67	0,0	$3{,}36 \times 10^{-2}$	$7{,}99 \times 10^{-3}$	0,55
Predicción	0,01	0,0	$4{,}6 \times 10^{-3}$	$7{,}99 \times 10^{-3}$	0,0

Tabla 2: Tiempo de Entrenamiento y de Predicción según cada modelo

La Figura 3 muestra la región de decisión de nuestro modelo y la trayectoria asociada:

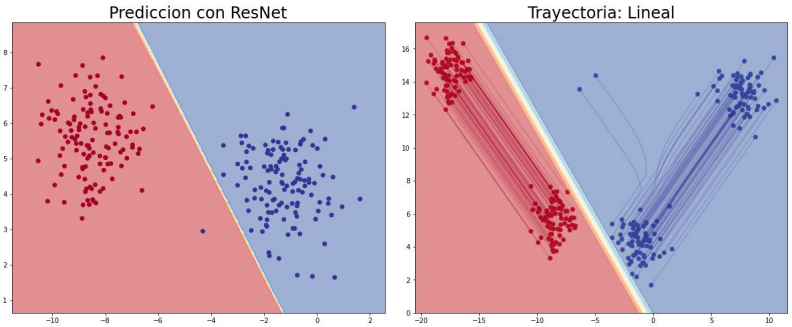

Figura 3: Región de Decisión obtenida por ResNet y Trayectoria asociada

Resultados: Set de Datos Donut

El control u óptimo encontrado logra que nuestro funcional J alcance un valor de $J = 4{,}79 \times 10^{-7}$.

La tabla 3 muestra las métricas obtenidas para cada modelo, mientras que la tabla 4 muestra los tiempos de ejecución asociados:

Modelo	**Correctitud**	**Sensibilidad**	**Especificidad**	**Precisión**	**F1-score**
ResNet	98,0 %	100 %	95,92 %	96,23 %	98,08 %
DCT	98,0 %	96,08 %	100 %	100 %	98,0 %
QDA	99,0 %	98,04 %	100 %	100 %	99,01 %
SVC	100 %	100 %	100 %	100 %	100 %
MLP	100 %	100 %	100 %	100 %	100 %

Tabla 3: Tabla de indicadores de cada Modelo

Tiempo [seg] de	ResNet	DCT	QDA	SVC	MLP
Entrenamiento	138,99	0,0	$8{,}59 \times 10^{-3}$	$5{,}21 \times 10^{-3}$	0,49
Predicción	$1{,}1 \times 10^{-2}$	0,0	0,0	0,0	0,0

Tabla 4: Tiempo de Entrenamiento y de Predicción según cada modelo

La Figura 4 muestra la región de decisión del modelo ResNet y la trayectoria asociada:

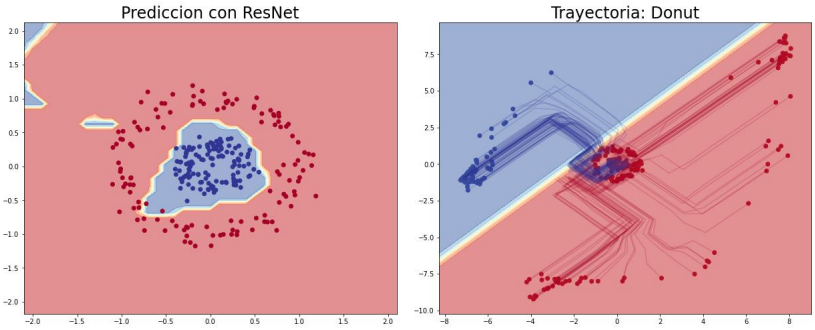

Figura 4: Región de Decisión obtenida por ResNet y Trayectoria asociada

Resultados: Set de Datos Spiral

El control u óptimo encontrado logra que nuestro funcional J alcance un valor de $J = 2{,}1 \times 10^{-7}$.

La tabla 5 muestra las métricas obtenidas para cada modelo, mientras que la tabla 6 muestra los tiempos de ejecución asociados:

Modelo	Correctitud	Sensibilidad	Especificidad	Precisión	F1-score
ResNet	99,0 %	98,08 %	100 %	100 %	99,03 %
DCT	99,0 %	98,08 %	100 %	100 %	99,03 %
QDA	88,0 %	82,69 %	93,75 %	93,48 %	87,76 %
SVC	99,0 %	98,08 %	100 %	100 %	99,03 %
MLP	100 %	100 %	100 %	100 %	100 %

Tabla 5: Tabla de indicadores de cada Modelo

Tiempo [seg] de	ResNet	DCT	QDA	SVC	MLP
Entrenamiento	164,84	$9{,}9 \times 10^{-4}$	$9{,}9 \times 10^{-4}$	$9{,}9 \times 10^{-4}$	0,56
Predicción	$9{,}9 \times 10^{-3}$	$9{,}7 \times 10^{-4}$	0,0	$9{,}8 \times 10^{-4}$	$9{,}9 \times 10^{-4}$

Tabla 6: Tiempo de Entrenamiento y de Predicción según cada modelo

La Figura 5 muestra la región de decisión del modelo ResNet y la trayectoria asociada:

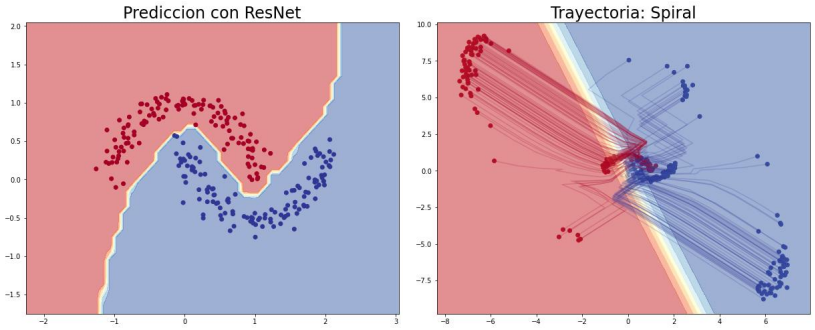

Figura 5: Región de Decisión obtenida por ResNet y Trayectoria asociada

Resultados: Set de Datos XOR

El control u óptimo encontrado logra que nuestro funcional J alcance un valor de $J = 3{,}31$.

La tabla 7 muestra las métricas obtenidas para cada modelo, mientras que la tabla 8 muestra los tiempos de ejecución asociados:

Modelo	**Correctitud**	**Sensibilidad**	**Especificidad**	**Precisión**	**F1-score**
ResNet	92,0 %	97,87 %	86,79 %	86,79 %	92,0 %
DCT	100 %	100 %	100 %	100 %	100 %
QDA	93,0 %	100 %	86,79 %	87,04 %	93,07 %
SVC	95,0 %	95,74 %	94,34 %	93,75 %	94,74 %
MLP	94,0 %	100 %	88,68 %	88,68 %	94,0 %

Tabla 7: Tabla de indicadores de cada Modelo

Tiempo [seg] de	**ResNet**	**DCT**	**QDA**	**SVC**	**MLP**
Entrenamiento	1327,49	$9{,}9 \times 10^{-4}$	$1{,}9 \times 10^{-3}$	$9{,}9 \times 10^{-4}$	0,5
Predicción	$1{,}6 \times 10^{-2}$	$9{,}9 \times 10^{-4}$	0,0	$1{,}9 \times 10^{-3}$	$1{,}0 \times 10^{-3}$

Tabla 8: Tiempo de Entrenamiento y de Predicción según cada modelo

La Figura 6 muestra la región de decisión del modelo ResNet y la trayectoria asociada:

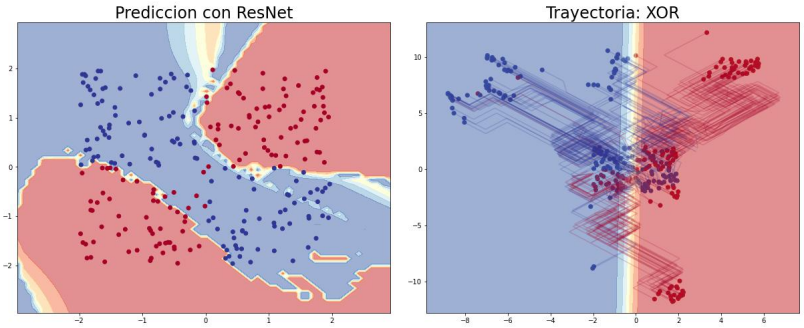

Figura 6: Región de Decisión obtenida por ResNet y Trayectoria asociada

5. Discusión y Conclusiones

En el transcurso de nuestro proyecto, abordamos la resolución de un problema de clasificación binaria mediante la aplicación de un enfoque basado en control óptimo. Utilizando un método numérico fundamentado en el método de Euler, logramos identificar, en diversos conjuntos de datos, un control óptimo que eficazmente minimiza el funcional objetivo J.

En particular, para el primer set de datos considerado, ResNet se destaca con una precisión del 99 % y una sensibilidad del 98,04 %, lo cual indica que falló en clasificar solo un par de los datos azules. A pesar de un tiempo de entrenamiento más prolongado (29,67 segundos) en comparación a los demás modelos, la velocidad de predicción es excepcionalmente rápida (0,01 segundos) al igual que los demás modelos (los cuales recordemos ya estan implementados y optimizados en la libreria correspondiente), remarcando la eficiencia de la red neuronal en la clasificación de un set de datos del tipo lineal.

Por otro lado, para el set de datos Donut, ResNet mantiene un rendimiento sólido, con una correctitud del 98 % y una sensibilidad del 100 %, aunque notamos que en especificidad es levemente inferior con un porcentaje de $95,92\,\%$, es decir hay datos rojos que clasifica como azules, esto es evidente al mirar la figura 4, vemos que el círculo que debiese formarse en el centro del gráfico de la región de decisión tiene una pequeña "deformidad"hacia un lado de la región roja. Nuevamente el tiempo de entrenamiento de nuestro modelo es relativamente alto (138,99 segundos), sobre todo al compararlo con los demás modelos.

Para el set de datos Spiral, vemos nuevamente que los indicadores tienen valores muy altos y se puede considerar que se mantiene a la par con los demás modelos, salvo con QDA, el cual está por debajo, es decir, nuestro modelo clasifica mejor los datos que el modelo QDA, aunque es sabido que dicho modelo no es precisamente el más óptimo para clasificar este tipo de datos. El tiempo de entrenamiento de ResNet (164.84 segundos) es más extenso que los demás tiempos, y es más extenso que en los tiempos considerados en los previos set de datos.

Para el último set de datos considerado XOR, ResNet exhibe un rendimiento competitivo con una correctitud del 92 % y una sensibilidad del 97,87 %, es decir, clasifica bien los azules, sin embargo la especificidad y la precisión tienen un valor de $86,79\,\%$, la cual sigue siendo alta, pero es lo más bajo hasta ahora si consideramos los demás set de datos, esto además muestra que hay una cantidad (relativamente no menor) de datos rojos que no está clasificando bien. Sin embargo notamos que los indicadores de todos los demás modelos decaen ante este set de datos, salvo por el árbol de decisión que tiene indicadores perfectos (aunque esto es esperable de dicho modelo dado este set de datos). El tiempo de entrenamiento de ResNet es sustancialmente más altos (1327,49 segundos), lo cual podría relacionarse con la complejidad inherente del conjunto de datos XOR.

ResNet, como se evidencia en los resultados, demuestra un rendimiento sólido en la tarea de clasificación. Las métricas generales revelan que ResNet es comparable, e incluso superior en algunos aspectos, a modelos tradicionales de clasificación. Aunque se observan tiempos de entrenamiento más largos en comparación a los otros modelos, la consistencia en las métricas de desempeño sugiere que la capacidad de aprendizaje de la red es sólida. Notamos además que a medida que los datos son más çomplejos"mayor es el tiempo de entrenamiento, esto se puede explicar debido a la implementación del modelo, puesto que es claro que la implementación se podría optimizar y lo estamos comparando con modelos que ya estan implementados y optimizados en librerias. Además de por sí las redes neuronales requieren de cierto tiempo de entrenamiento, tan solo basta mirar el tiempo que tarda MLP, en todos los set de datos tarda alrededor de medio segundo, sin embargo los demás modelos tardan del orden de 10^{-3} o 10^{-4} segundos.

Cabe destacar además que las figuras relacionadas a la trayectoria asociada al control óptimo de cada problema (es decir, del problema de minimización asociado a cada set de datos) nos ofrece una visualización de lo que hace el modelo. En las capas internas se trasladan los datos, con tal de dejarlos separados según su clasificación, y en la capa de salida (es decir, cuando se aplica $C(W \cdot + \mu)$), se clasifica dependiendo del lado en el que se hayan quedado según la traslación, por ejemplo, para el set de datos lineal, es claro que los datos ya estan separados, pero nuestro modelo los separa aún mas y luego hace la clasificación correspondiente, o por ejemplo para el set de datos Spiral, toma los datos de abajo (los azules) y los traslada hacia la derecha, mientras que a los de arriba hacia la izquierda.

Para concluir, notamos que los resultados respaldan la efectividad de nuestro enfoque de control óptimo para abordar problemas de clasificación. La capacidad de ResNet para manejar conjuntos de datos desafiantes y su eficiencia computacional destacan su potencial para contribuir significativamente a desarrollos futuros en esta área.

6. Bibliografía

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778)

[2] Benning, M., Celledoni, E., Ehrhardt, M. J., Owren, B., & Schönlieb, C. B. (2019). Deep learning as optimal control problems: Models and numerical methods.

7. Anexo

Las Figuras 7,8,9,10 muestran las Regiones de Decisión obtenidas según los datasets Lineal, Donut, Spiral y XOR respectivamente, para los distintos modelos usados para comparar:

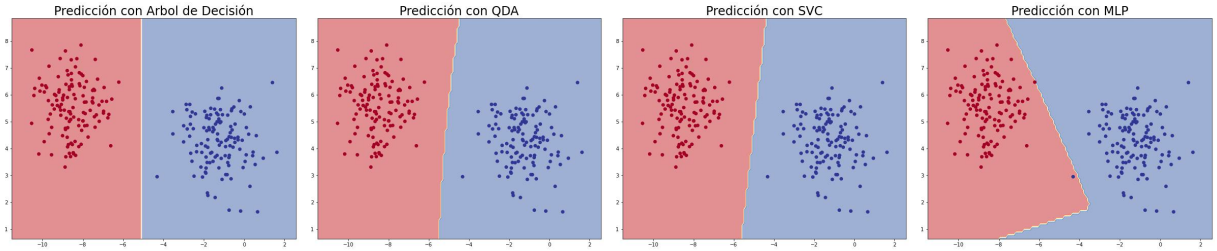

Figura 7: Regiones de Decisión obtenidas según cada modelo considerando el set de datos Lineal

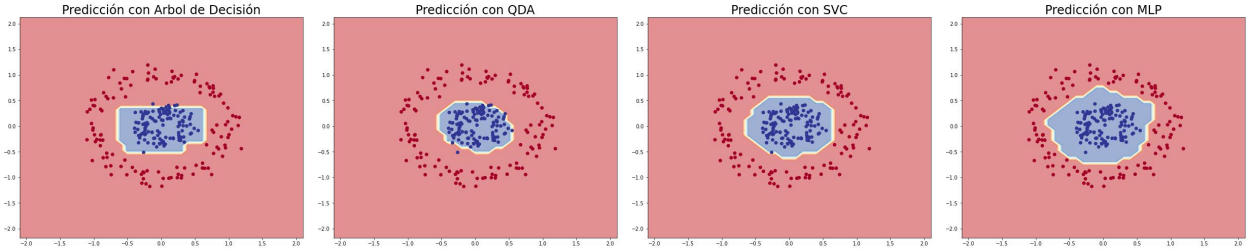

Figura 8: Regiones de Decisión obtenidas según cada modelo considerando el set de datos Donut

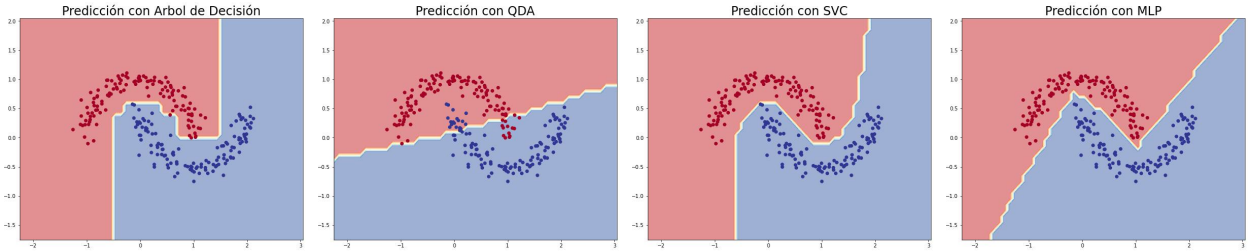

Figura 9: Regiones de Decisión obtenidas según cada modelo considerando el set de datos Spiral

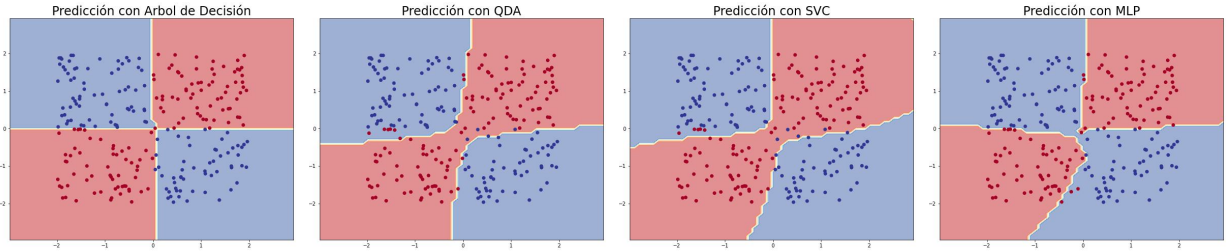

Figura 10: Regiones de Decisión obtenidas según cada modelo considerando el set de datos XOR