Entendiendo Aprendizaje Profundo como un Problema de Control Óptimo

Felipe Urrutia, Daniel Minaya, Felipe Rivera

Ingeniería Matemática
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

furrutiav/
deep-learning-as-optimal-control

MA4703
Control Óptimo: Teoría y Laboratorio

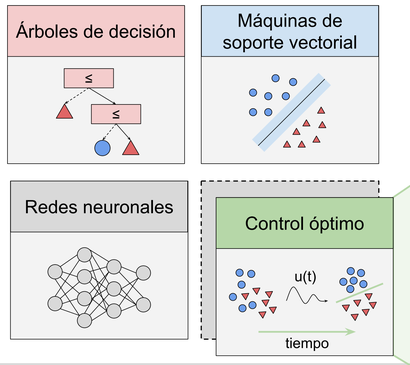
Motivación

Problema de clasificación binaria

El problema de clasificación es una de las tareas más estudiadas en el área de aprendizaje de máquinas [1]. Modelos clásicos, algunos interpretables (árboles de decisión) y otros estables (máquinas de soporte vectorial), logran un desempeño inferior a los modelos cajas negra basados en aprendizajes profundo sin garantías de estabilidad [2]. Por esto, se estudia el problema de clasificación como un problema de control óptimo, para encontrar modelos con las mejores propiedades.

Pregunta del proyecto

¿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?



Introducción

Modelo ResNet [2]

Queremos resolver un problema clasificación binarias como un problema de control óptimo tipo Mayer dado como sigue

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

Se realiza una discretización tipo Euler para la dinámica del problema. De esta manera el control viene dado por

$$u = \left(u^{[0]}, \dots, u^{[N-1]}\right), \ u^{[j]} = \left(K^{[j]}, \beta^{[j]}\right)$$

Mientras que la variable estado del problema se describe por

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

De esta manera, la dinámica discretizada sigue el siguiente esquema numérico

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

donde σ es la función de activación tangente hiperbólica.

Principio de Pontryagin (PP)

$$\min_{u(\cdot)} \ \mathcal{J}(y(T))$$
$$\text{s.a. } \dot{y} = f(y,u) \text{ en } [0,T]$$
$$y(0) = x$$

Hamiltoniano

$$H(y,p,u) = p^T f(y,u) = p^T \sigma(Ky+\beta)$$
$$\mathcal{J}(z) = \tfrac{1}{2}|\mathcal{C}(Wz+\mu) - c|^2$$
$$u = (K, \beta)$$

Sistema del estado adjunto

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

Condición de transversalidad

$$dg = \partial_y g(y_f) dy_f, \ [\Theta(t)dt - p(t)^T dy_t]_{t=0}^{t=t_f} = -p(t_f)^T dy_f \quad p(T) = \frac{\partial \mathcal{J}}{\partial y}(y(T))$$

Ecuación de Hamilton-Jacobi-Bellman (HJB)

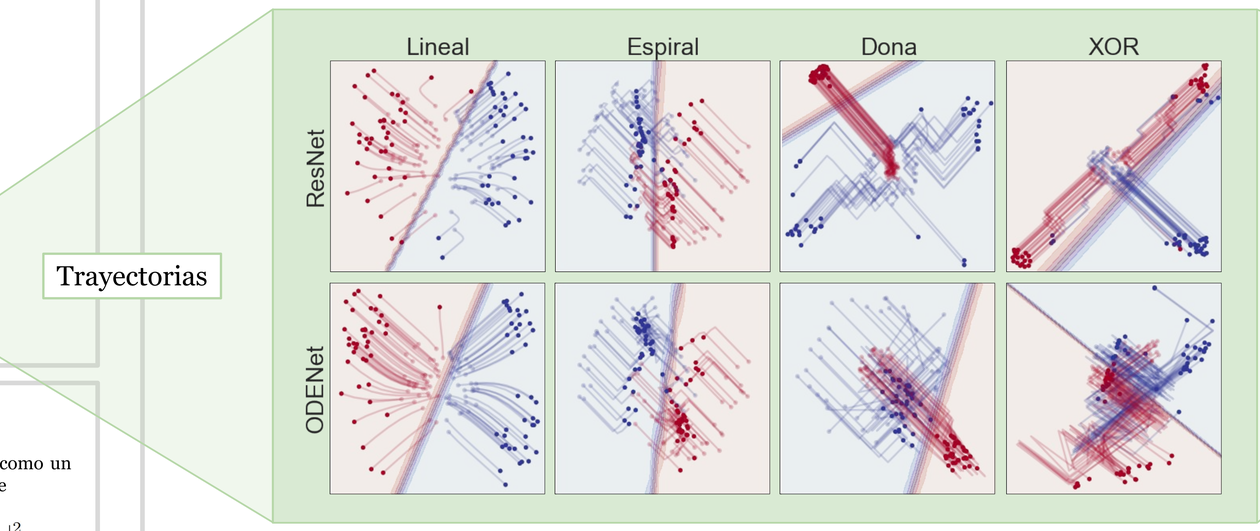
$$(P_{t_0,y_0}) \quad \min_{u(\cdot)} \ J_B = \int_{t_0}^{T} e^{-\lambda t} \ell(y(t), u(t)) dt + \mathcal{J}(y(T))$$
$$\text{s.a. } \dot{y} = f(y,u) \text{ en } [t_0, T]$$
$$y(t_0) = y_0$$

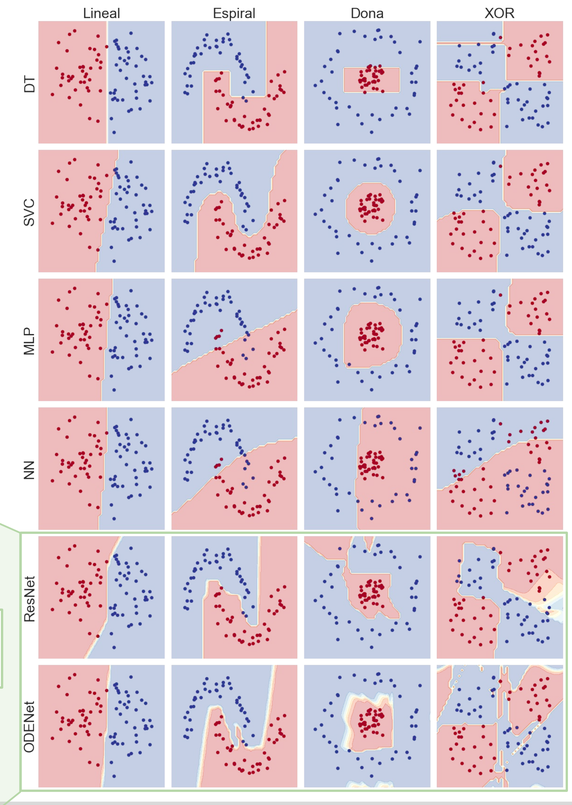
En nuestro problema $\ell \equiv 0$ y $\lambda = 0$. De esta manera, obtenemos que si consideramos $V(t_0, y_0)$ como la función valor de (P_{t_0,y_0}), entonces

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

Resultados teóricos

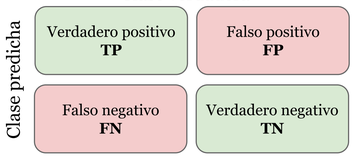
Resultados numéricos





Desempeño

Dataset	Lineal		Espiral		Dona		XOR	
Conjunto	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



$$\frac{TP+TN}{TP+FP+TN+FN}$$

Desempeño y complejidad de los modelos

Train (s)
Tiempo utilizado para entrenar el modelo

Test (ms)
Tiempo utilizado por el modelo para realizar la inferencia (o predicción)

Complejidad

Dataset	Lineal		Espiral		Dona		XOR	
Tiempo	Train (s)	Test (ms)	Train (s)	Test (ms)	Train (s)	Test (ms)	Train (s)	Test (ms)
DT	0.001	0.000	0.001	0.000	0.000	0.000	0.001	0.000
SVC	0.000	0.000	0.001	1.000	0.000	0.000	0.001	0.000
MLP	0.088	0.000	0.092	0.000	0.087	0.000	0.091	0.000
NN	0.136	0.000	0.137	0.000	0.139	0.000	0.139	0.000
ResNet	33.598	10.010	57.725	10.009	184.400	11.010	430.618	11.010
ODENet	6.501	5.004	29.140	4.004	28.635	4.003	62.311	4.003

Conclusiones

Resultados Teóricos
- ❖ PP y HJB, nos asegura la existencia de un control óptimo bajo ciertas condiciones
- ❖ Mayor interpretabilidad del control obtenido y entender la naturaleza de los datos

Resultados Numéricos
- ❖ ResNet y ODENet, evidencian cómo producen la salida a partir de trayectorias
- ❖ Logran un desempeño competitivo con respecto a los modelos de referencia
- ❖ Las implementaciones pueden ser optimizadas para mejorar la complejidad e incluso el desempeño

Trabajo Futuro
- ➔ Estudiar el problema de control óptimo con tiempo final libre y/o penalizar la energía de la dinámica
- ➔ Estudiar dinámicas en derivadas parciales y su similitud con los modelos de redes neuronales de convolución [3]

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*.

[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.