

Intro. a redes neuronales

Martín Quijano - Martina Coletto

Dataset 1 - Media Luna Abstracta

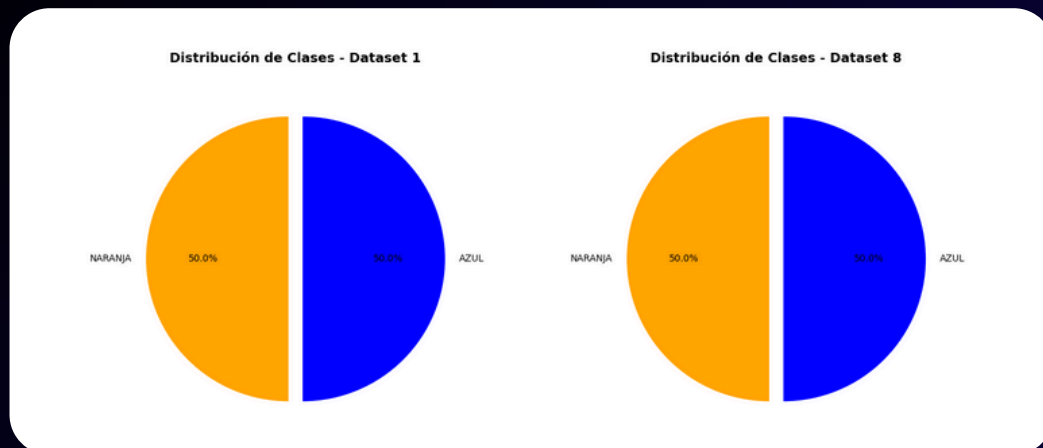
- Contiene puntos distribuidos en dos medias lunas entrelazadas.
- Cada punto tiene coordenadas X_1 , X_2 y una clase: NARANJA o AZUL.
- Presenta una frontera no lineal, lo que lo vuelve un buen caso para redes neuronales con funciones de activación no lineales.
- Ideal para evaluar la capacidad del modelo de separar clases en formas complejas.
- No hay overlapping entre los puntos y color

Dataset 8 - Corazón Dividido

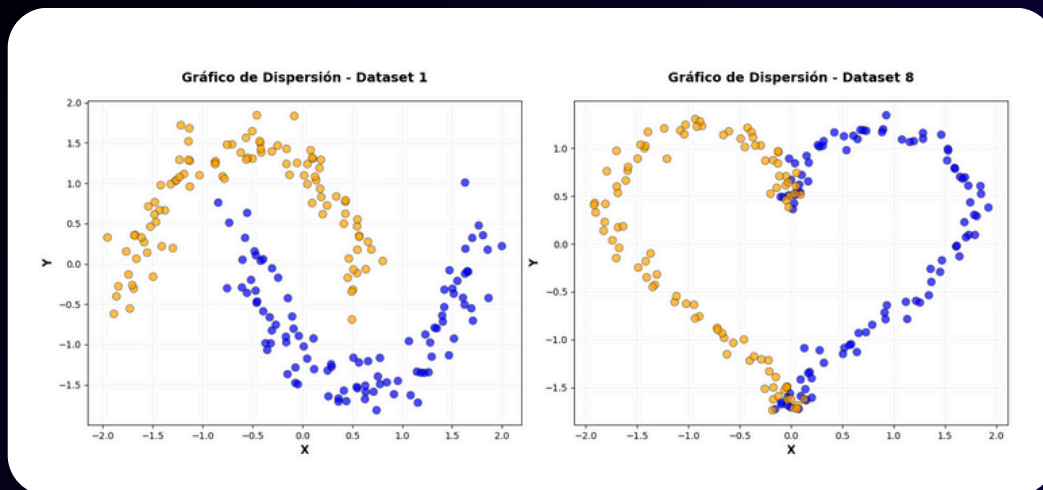
- Los puntos están distribuidos en forma de corazón, dividido en dos clases.
- Cada observación incluye X_1 , X_2 y la clase correspondiente.
- Requiere que el modelo se adapte a una forma cerrada y simétrica.
- Permite analizar cómo la red neuronal traza fronteras de decisión en regiones curvas.
- Hay overlapping entre los puntos y color



Análisis exploratorio



Para ambos datasets hicimos una visualización inicial de los datos. verificamos que ambas clases estuvieran balanceadas



observar la distribución espacial de las clases y confirmar que se trataba de casos no lineales.

El primer dataset que analizamos fue el de Media Luna Abstracta. Tiene puntos distribuidos en dos medias lunas entrelazadas, con coordenadas X1 y X2, y una etiqueta de clase: Naranja o Azul.

El segundo dataset tiene una distribución en forma de corazón, también con dos clases.

Lo importante a destacar es que en el dataset1 los puntos estan claramente separados, mientras que en el dataset 8 hay overlapping.



Arquitectura de la red neuronal

Al tener formas distintas, decidimos usar arquitecturas de redes distintas

Modelo feedforward

Modelo de red neuronal feedforward diseñado para clasificación binaria

Usamos un modelo feedforward con capas ocultas con activación ReLU, y una capa de salida con activación Sigmoid para clasificación binaria.

Arquitectura detallada Dataset 1

| capa | tipo | neuronas | activación |
|-----------------|-------|----------|------------|
| entrada | - | 2 (x, y) | - |
| 7 capas ocultas | dense | 10 | ReLu |
| capa de salida | dense | 1 | Sigmoid |

Entrada: Contamos con dos neuronas que representan nuestras variables (x, y). Esta es la base de nuestro modelo, donde comenzamos a procesar la información.

Capas ocultas: tenemos 7 capas densas, cada una con 10 neuronas. Utilizamos la función de activación ReLu, que es para introducir no linealidades en nuestro modelo. Esto permite a la red aprender patrones más complejos.

Salida: nuestra capa de salida es también densa, con una única neurona que utiliza la activación Sigmoid. Esto nos ayudará a clasificar nuestras entradas en dos clases: NARANJA o AZUL.

Arquitectura detallada Dataset 8

| capa | tipo | neuronas | activación |
|-----------------|-------|----------|------------|
| entrada | - | 2 (x, y) | - |
| 2 capas ocultas | dense | 8 | ReLu |
| capa de salida | dense | 1 | Sigmoid |

Entrada: Contamos con dos neuronas que representan nuestras variables (x, y). Esta es la base de nuestro modelo, donde comenzamos a procesar la información.

Capas ocultas: tenemos 3 capas densas, cada una con 10 neuronas. Utilizamos la función de activación ReLu, que es para introducir no linealidades en nuestro modelo. Esto permite a la red aprender patrones más complejos.

Salida: nuestra capa de salida es también densa, con una única neurona que utiliza la activación Sigmoid. Esto nos ayudará a clasificar nuestras entradas en dos clases: NARANJA o AZUL.

Parámetros de entrenamiento

- Función de pérdida: `binary_crossentropy`
- Optimizador: Adam
- Épocas: 100
- Tamaño del batch: 32
- Validación: cross validation con 5 folds

Los modelos se entrenaron con la función de pérdida `binary_crossentropy`, usando el optimizador Adam durante 100 épocas y con un batch size de 32. También aplicamos cross validations con 5 folds para no sesgar la muestra y aplicar el modelo final a la muestra completa

Objetivo

Aprender a clasificar puntos 2D (X, Y) en dos clases (NARANJA o AZUL), adaptándose a fronteras no lineales.

El objetivo era que el modelo aprendiera a clasificar puntos en 2D adaptándose a las fronteras no lineales.



Métricas de Evaluación

Evaluación en test set

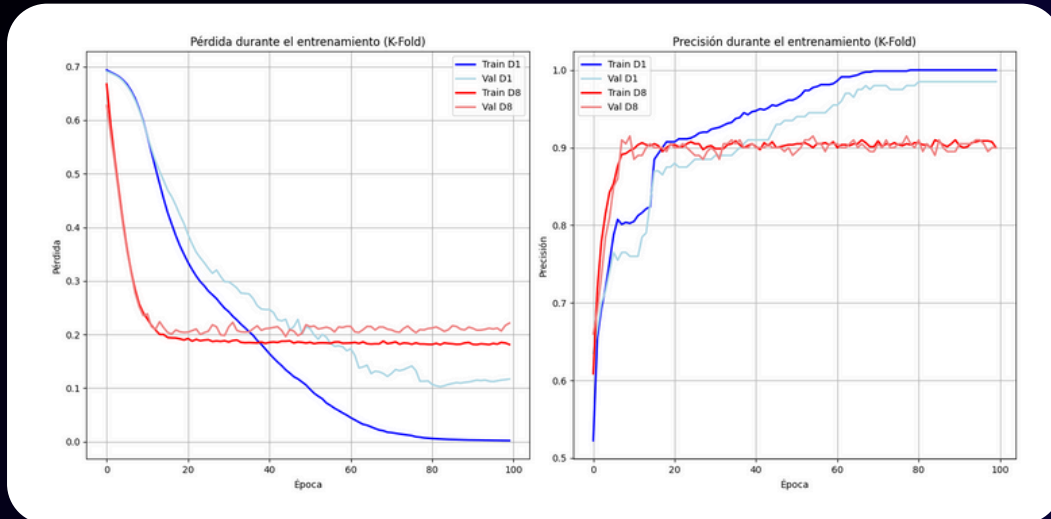
- Clasificación binaria (0 = NARANJA, 1 = AZUL)

| Dataset | Accuracy | Precisión | Recall | F1-score |
|------------------------------|----------|-----------|--------|----------|
| Dataset 1 (Media Luna) | 1.000 | 1.000 | 1.000 | 1.000 |
| Dataset 8 (Corazón Dividido) | 0.9100 | 0.9100 | 0.9100 | 0.9100 |

“Acá vemos los resultados sobre el test set. Para Media Luna obtuvimos un accuracy de 87.5%, y para Corazón Dividido, un 95%. La precisión, recall y F1-score también fueron superiores en el segundo dataset, lo que sugiere que la forma del corazón fue más fácil de aprender para la red.”



Gráficos de entrenamiento



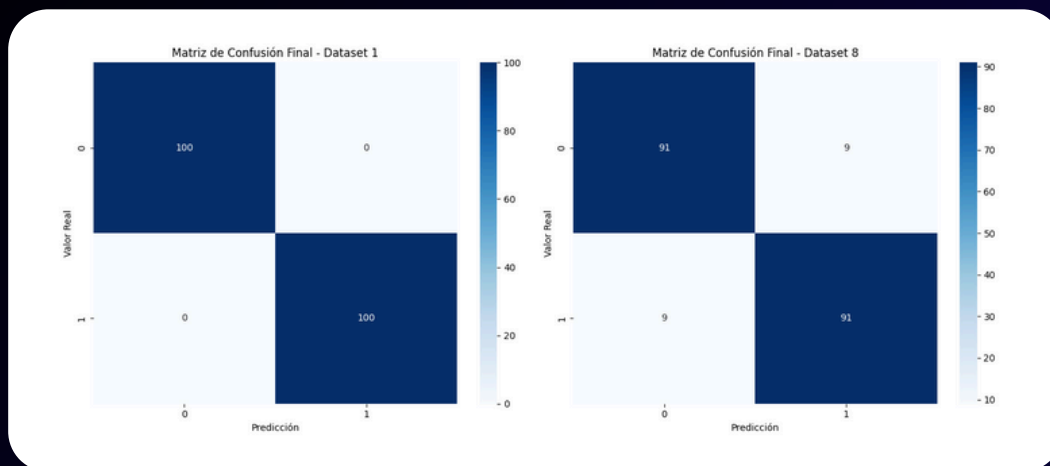
Estos son los gráficos de pérdida y accuracy a lo largo de las épocas.

Se puede observar una convergencia clara en ambos casos, a partir de la epoca 60.

Vemos como el train D1 tiene una loss de casi 0 y una precision de casi 1, mientras que las curvas del d8 no llegan a valores tan buenos.

Tambien nos parece interesante ver como en loss las curvas del D8 convergen con menor cantidad de epocas que las curvas de D1.

Tambien es interesante ver como las curvas de precision convergen con menor cantidad de epoca que loss.



Estas son las matrices de confusión para cada modelo.

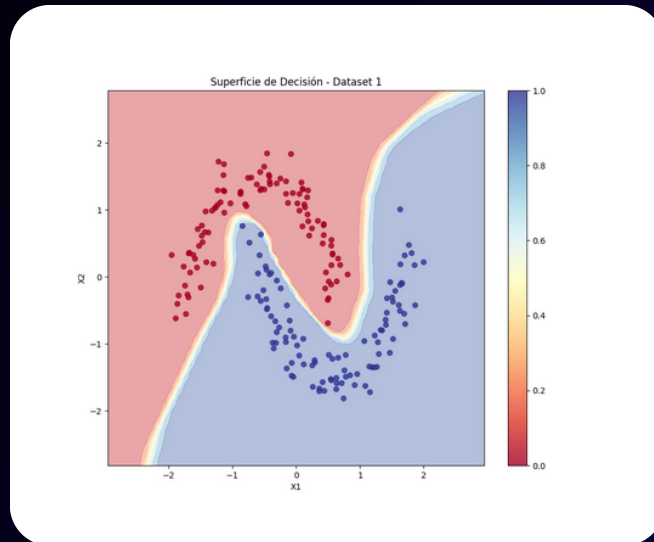
En ambos casos se ve un buen nivel de predicción, con la mayoría de los puntos correctamente clasificados.

Para Media Luna hay solamente un error, que es un punto que se ve en un siguiente grafico.

En cambio, el modelo para Corazón Dividido muestra que hay mucho mas errores, especialmente cuando se predice azul pero es naranja, como vamos a ver ahora.

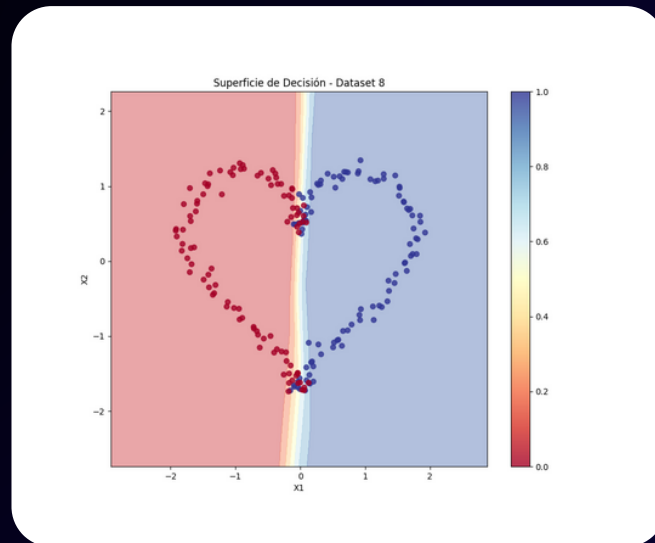


Gráficos de fronteras de decisión



Aca vemos el decision boundary para el dataset 1. Como se ve, estan todos los puntos bien clasificados excepto por ese punto rojo en el area azul.

Vemos tambien que los puntos donde el modelo predice valores cercanos a 0.5 (zonas blancas) es bastante estrecha. Se podrian sumar neuronas y/o capas para reducir esta zona.



En el dataset del corazón, la frontera se tira hacia la izquierda, causando los errores que veíamos previamente. Intentamos cambiar la cantidad de neuronas y capas pero no pudimos mejorar mas que esto el modelo, incluso usando la tecnica de regularizacion L2.

Aun asi, si el mejor modelo posible para este dataset no llegaria a ser tan bueno como nuestro modelo del dataset 1, y es que en el modelo anterior si bien tenemos una forma no lineal compleja, los datos tienen separacion, lo que nos permite agregar capas ocultas para generar esa forma y poder hacer predicciones correctas. Por otro lado, este otro dataset tiene una forma un poco mas simple pero con datos overlappeados, lo que hace imposible generar un modelo perfecto.

Gracias

Martín Quijano - Martina Coletto