

Universidad San Francisco de Quito
Inteligencia Artificial

Nombre: José Gabriel García

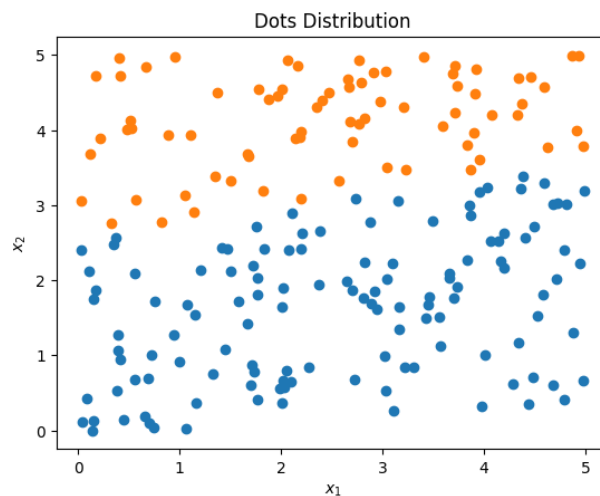
Código: 00211322

NRC: 3322

Tarea 3

Ejercicio 1: Puntos separados por una línea

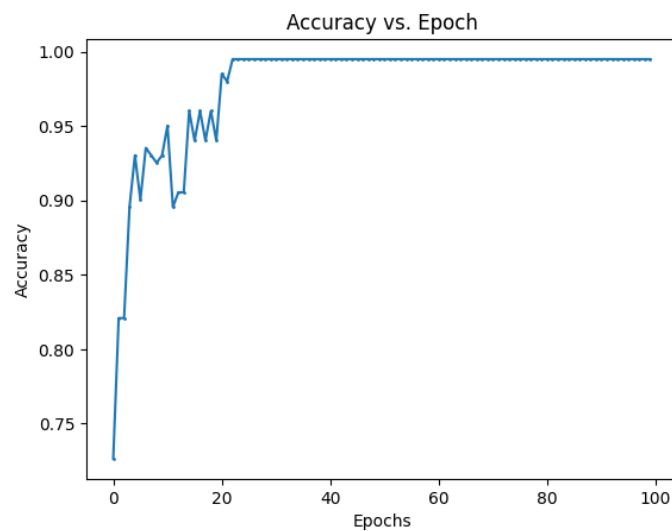
Para este ejercicio se cuenta con dos data sets: uno con puntos coordinados en 2D y otro con los labels de cada punto (1, -1). Dichos puntos marcados con diferentes colores para cada label se muestran a continuación.



De la imagen anterior, se puede observar que estos dos datasets son claramente separables por una línea. Para cada código, se utilizan 100 épocas y el mismo funcionamiento del perceptrón.

Implementación de Algoritmo de Aprendizaje En Secuencia

El Aprendizaje en Secuencia actualiza los pesos del perceptrón cada vez que evalúa un punto con los pesos anteriores. Así, la precisión del modelo por época se puede ver a continuación.



En la gráfica anterior se puede observar que en la primera época se tuvo una precisión del 73%, y este valor mejora hasta la época 30, para luego mantenerse en un valor constante. De este ejercicio se puede ver como es importante implementar un método que detenga el aprendizaje cuando la precisión deja de variar, o varía muy poco por algunas épocas.

El accuracy final del modelo fue del 99.5%. Este resultado es bastante bueno, pero se debe tener en cuenta que los datos son claramente separables por una línea por lo que es un resultado esperado. Además de esto, los pesos obtenidos fueron los siguientes:

w_0	w_1	w_2
-22	-1.54	8.16

Con estos pesos se puede encontrar la ecuación de la recta que divide los puntos. Para encontrar esta recta se utiliza la ecuación del modelo, la cual está dada por:

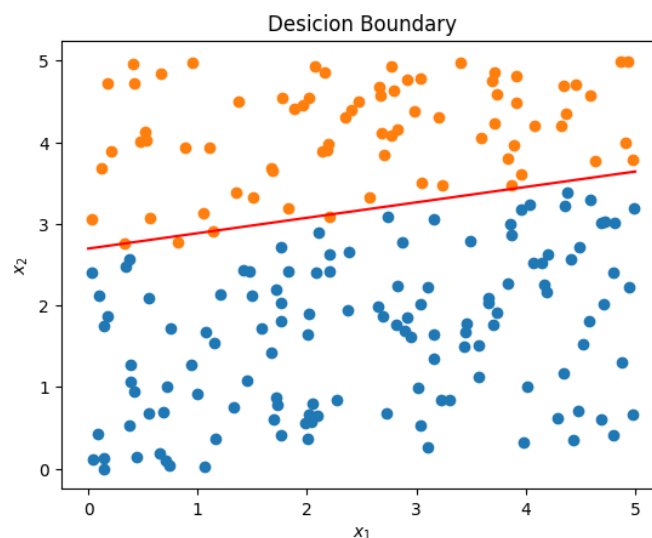
$$w_0 + w_1x_1 + w_2x_2 = y$$

Ya que los puntos se encuentran en el plano, entonces el valor de y debe tomar el valor de 0. Así, se tiene una ecuación en 2 dimensiones y al despejar x_2 se encuentra se puede encontrar la ecuación de dicha recta.

$$x_2 = \frac{-w_0 - w_1x_1}{w_2}$$

$$x_2 = 2.696 + 0.189x_1$$

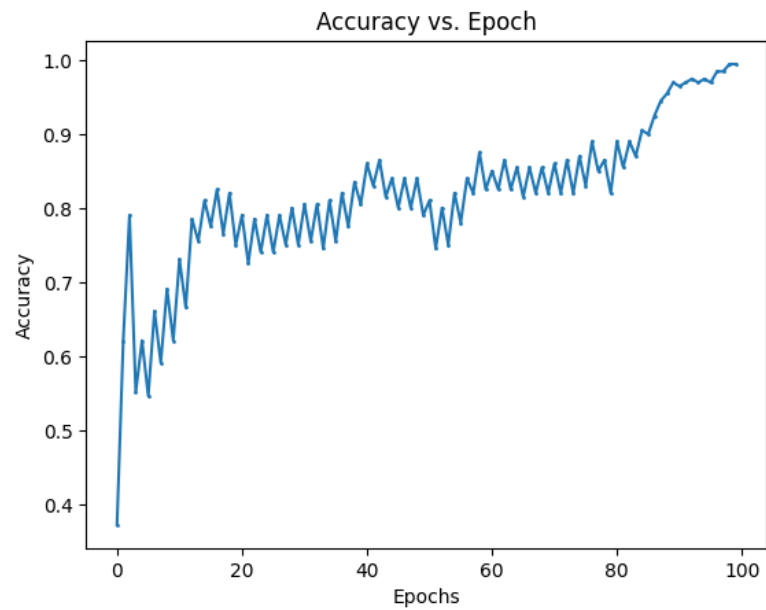
Así, al graficar esta línea se obtiene lo siguiente:



En donde se puede ver como el resultado obtenido es bastante bueno, y separa muy bien a las dos clases que se tiene.

Implementación de Algoritmo de Aprendizaje En Batch

En este algoritmo se actualizan todos los pesos una vez después de que todos los puntos del dataset han sido predichos por un arreglo de pesos anterior. La precisión de este modelo por época se muestra a continuación.



El primer valor de accuracy obtenido es del 38%, y luego este empieza fluctuar e incrementarse de manera lenta hasta alcanzar una precisión del 99.5%.

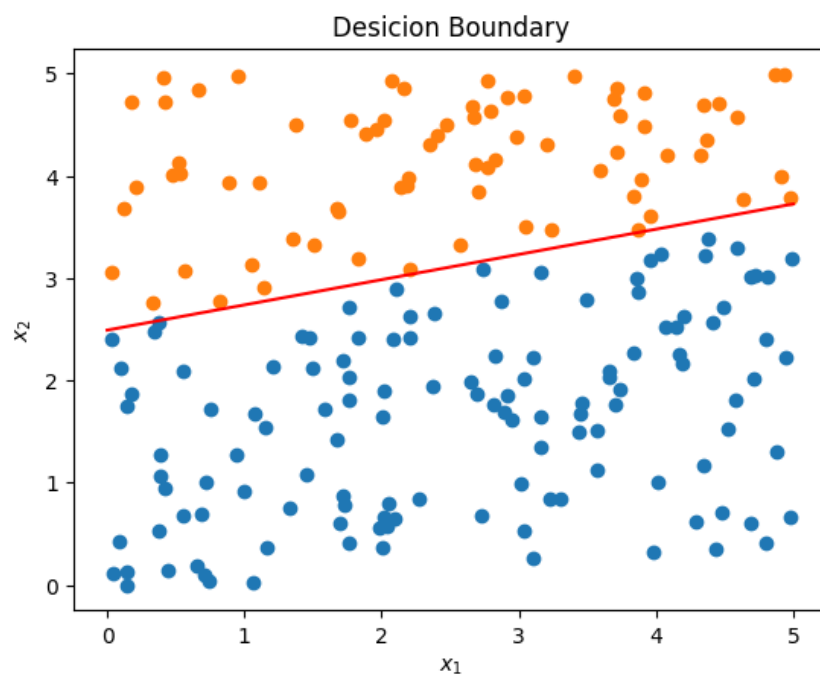
Los pesos obtenidos por este algoritmo son los siguientes:

w_0	w_1	w_2
-329.2	-32.62	132.18

Y, siguiendo el mismo procedimiento anterior para obtener la recta que separa los puntos, se obtiene lo siguiente:

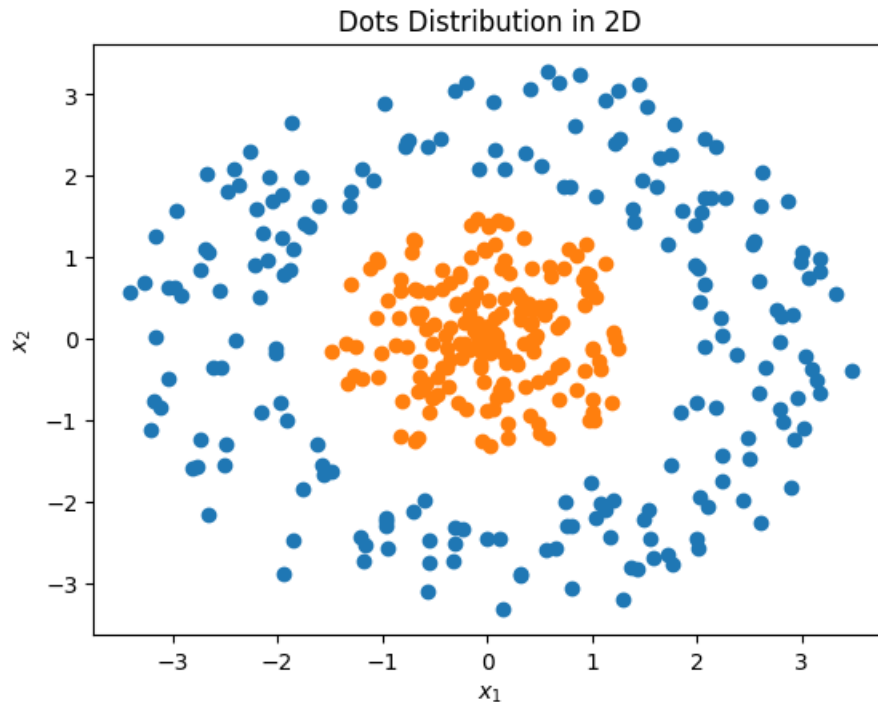
$$x_2 = 2.491 + 0.247x_1$$

Y la línea que se obtiene es la siguiente:



Ejercicio 2: Puntos con 2 regiones diferentes: una dentro de la otra

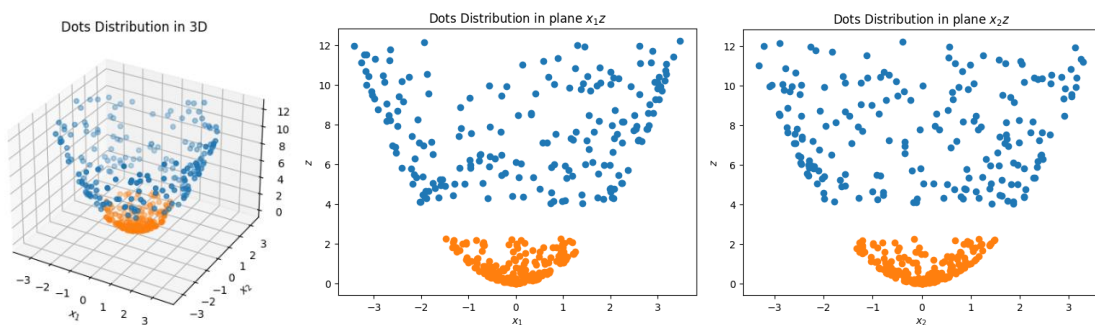
Este ejercicio cuenta con 2 datasets: uno con las coordenadas de varios puntos, y otro con los labels de cada punto. A continuación, se muestra este data set.



De la imagen anterior, se puede observar como los puntos no son separables por una línea recta de ninguna forma, por lo que una alternativa es aumentar la dimensionalidad de los datos para tratar de separarlos por un plano. Así, se utilizan la siguiente ecuación para determinar la tercera coordenada de los puntos:

$$z = x_1^2 + x_2^2$$

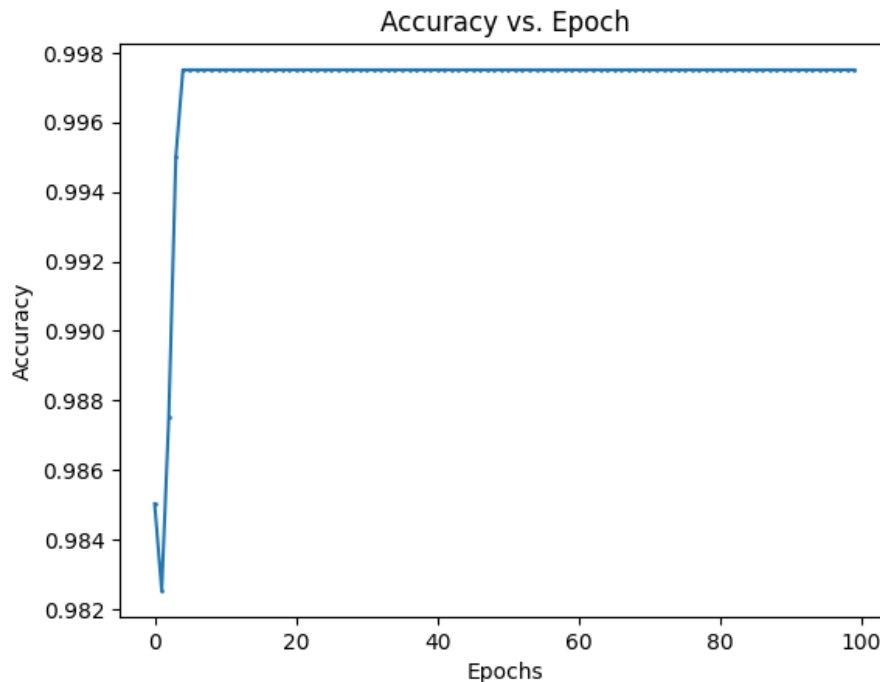
Aumentar la dimensionalidad de los datos es una alternativa que puede utilizarse siempre para separar los datos por un plano (3D) o un hiperplano en más de 3 dimensiones. Así, al obtener esta tercera dimensión y graficar los datos se obtuvo lo siguiente:



De las gráficas anteriores, se puede observar como en 3D los puntos si son separables por un plano. Así, con el código implementado se utilizarán 100 épocas.

Implementación de Algoritmo de Aprendizaje En Secuencia

Este algoritmo arrojo el siguiente gráfico que muestra su mejoramiento por época.



El primer valor de accuracy de este modelo fue 98.5% el cual es un resultado sorprendente ya que la convergencia del modelo es bastante rápida lo que se puede deber a la nueva característica que existe, o que la inicialización de los pesos fue bastante buena; sin embargo, esto se puede descartar ya que al correr el algoritmo varias veces se obtuvieron resultados similares. Por otro lado, el valor de accuracy final fue de 99.75% el cuál es muy bueno y en la gráfica se puede observar que el mismo ya fue alcanzado alrededor de la época 10.

Los pesos encontrados fueron los siguientes:

w_0	w_1	w_2	w_3
3.4	-0.33	0.40	-1.37

Con estos pesos se puede encontrar la ecuación del plano que divide los puntos. Para encontrar este plano se utiliza la ecuación del modelo, la cual está dada por:

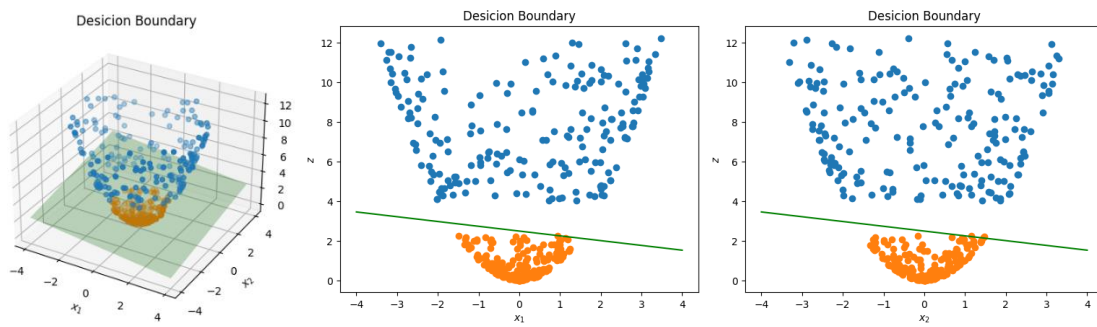
$$w_0 + w_1x_1 + w_2x_2 + w_3x_3 = y$$

La ecuación anterior es de un hiperplano, y ya que los datos se encuentran en 3D entonces se debe proyectar este hiperplano a las 3 dimensiones, y para hacerlo el valor de y debe ser cero.

$$x_3 = \frac{-w_0 - w_1x_1 - w_2x_2}{w_3}$$

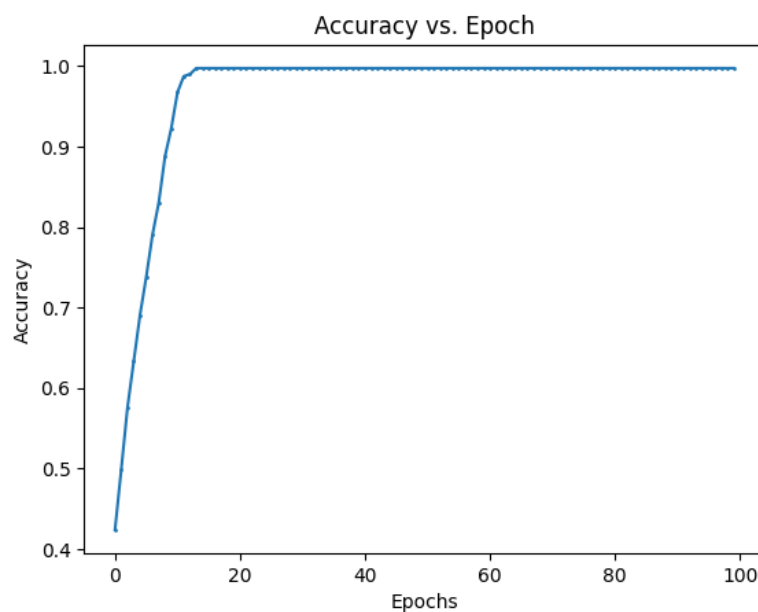
$$x_3 = 2.482 - 0.241x_1 + 0.292x_2$$

Y el plano que se obtiene es el siguiente:



Implementación de Algoritmo de Aprendizaje En Batch

El gráfico de mejoramiento por época de este modelo se muestra a continuación:



El primer valor de accuracy obtenido es de alrededor de 42%, y luego este valor aumenta hasta llegar a un final de 99.75% el cual es el mismo valor del algoritmo anterior. Además, se puede observar como el valor de accuracy final ya es alcanzado por el modelo en la época 20.

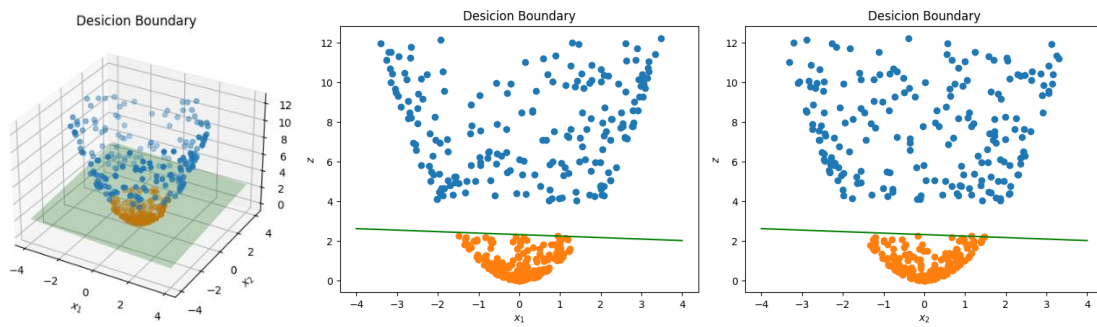
Los pesos obtenidos para este algoritmo fueron:

w_0	w_1	w_2	w_3
327	-10.69	21.14	-142.08

Y, siguiendo el mismo procedimiento anterior para obtener el plano que separa los puntos, se obtiene lo siguiente:

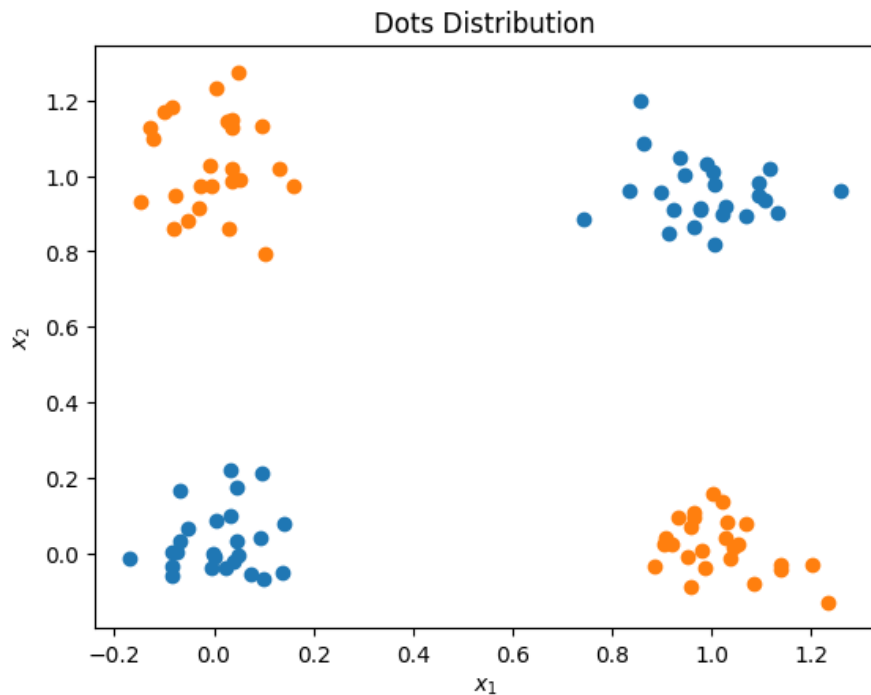
$$x_3 = 2.302 - 0.075x_1 + 0.149x_2$$

Y el plano que se obtiene es el siguiente:



Ejercicio 3: Problema XOR

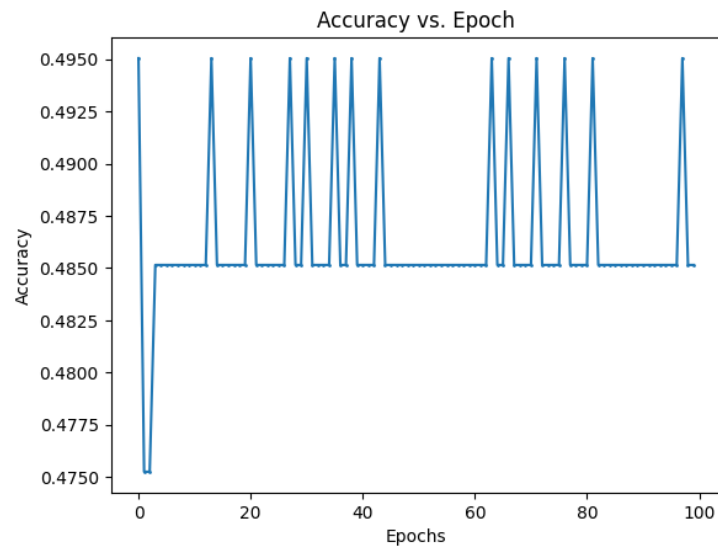
Como se vio en los ejercicios anteriores, varios problemas para separar sets de datos pueden solucionarse con un hiperplano, sin embargo, existen otros que no, y el problema de XOR es uno de ellos.



Como se puede ver en el gráfico anterior, es imposible separar los dos sets con una línea recta y los resultados que se obtienen al intentar estos datos con el perceptrón muestran a continuación.

Implementación de Algoritmo de Aprendizaje En Secuencia

El gráfico de la evolución del accuracy por época es el siguiente:



Como se puede observar en el gráfico anterior, el primer valor de accuracy es de 49.5%, luego baja a 47.5% y finalmente empieza a variar entre 48.5% y 49.5%. Como se puede observar, este modelo se adapta a los datos, y en caso de que se desee predecir un input al sistema, la probabilidad de que este acierta es menor que la de lanzar una moneda.

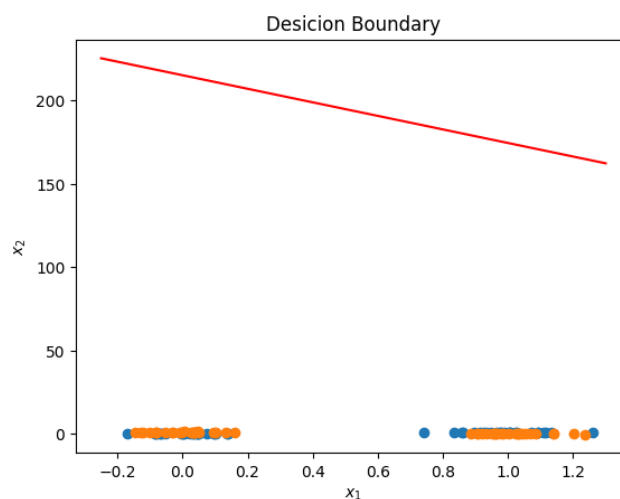
Los pesos encontrados fueron:

w_0	w_1	w_2
-0.8	0.15	0.0037

Y, siguiendo el procedimiento del ejercicio 1, la ecuación de la recta que mejor separa los datos según el modelo de ML usado sería:

$$x_2 = 216.216 - 40.541x_1$$

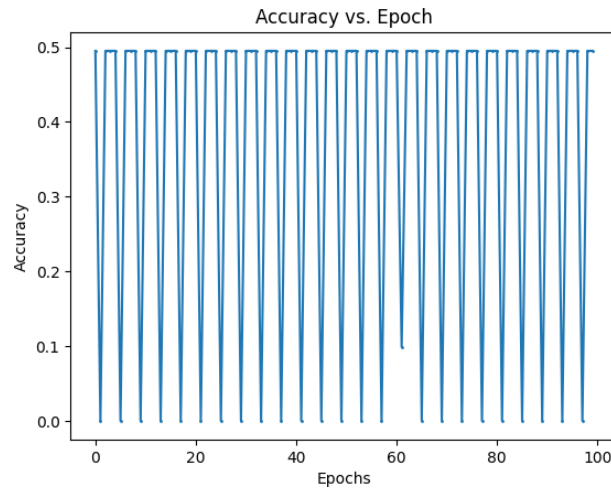
Y la línea recta obtenida:



En el gráfico anterior, se ve como el modelo genera una línea que ni siquiera pasa cerca de los datos, pero el accuracy que obtiene es el mejor ya que la mitad de los datos es correcta y la otra no.

Implementación de Algoritmo de Aprendizaje En Batch

La evolución del accuracy por época del modelo se muestra a continuación.



Como se puede observar en el gráfico anterior, el mejor accuracy que alcanza el modelo es casi el 50%, y el peor es de 0%, y durante todas las épocas este se mantiene variando entre estos dos valores. Al igual que con el modelo anterior, si se desea predecir un input a este modelo, da igual utilizar el modelo o lanzar una moneda.

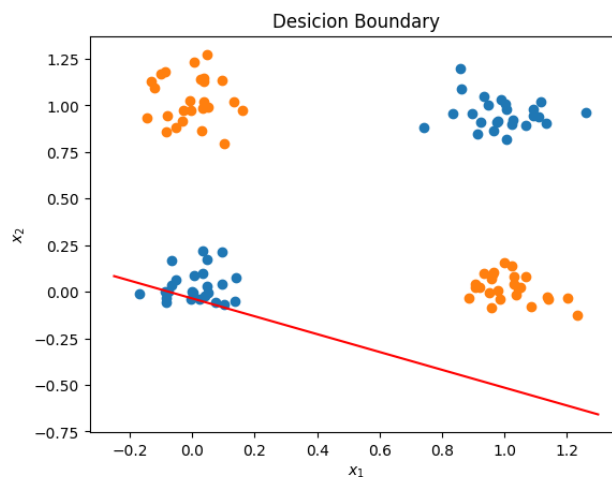
Los pesos encontrados fueron:

w_0	w_1	w_2
0.4	5.37	11.21

Y, siguiendo el procedimiento del ejercicio 1, la ecuación de la recta que mejor separa los datos según el modelo de ML usado sería:

$$x_2 = -0.0357 - 0.479x_1$$

Y la línea recta obtenida:



De la imagen anterior se observa que la línea que grafica encuentro el modelo de ML no separa los datos, además de que el resultado que brinda es bastante malo. Esta línea dibuja dos regiones, una sobre ella y otra bajo la misma, sin embargo, la mayoría de puntos están sobre la línea y es por eso que el modelo tiene un 50% de accuracy, ya que la mitad de puntos están bien clasificados y el resto no.

Para este ejercicio se podría intentar aumentar la dimensionalidad con funciones no lineales, pero este es un trabajo complicado que requiere de prueba-error, y de experiencia. Para este ejercicio se intentó usar las siguientes funciones:

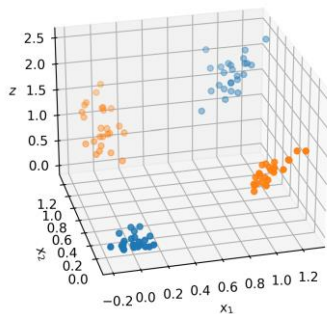
$$z = x_1^2 + x_2^2$$

$$z = \sin(x_1) + \cos(x_2)$$

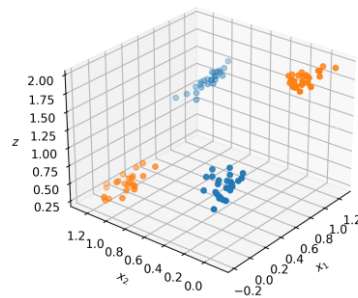
$$z = (\sin(x_1))^2 + \cos(x_2)$$

Y los resultados fueron los siguientes

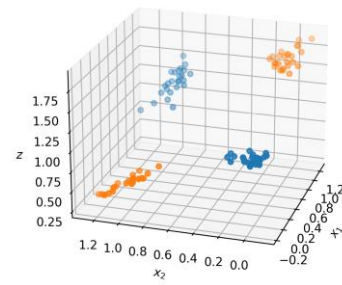
Dots Distribution in 3D with x_1^2 and x_2^2 function



Dots Distribution in 3D with \sin and \cos function



Dots Distribution in 3D with \sin^2 and \cos function



En las gráficas anteriores se puede observar que aunque se aumente una dimensión es imposible encontrar la forma de separar los dos set de datos con un plano, por lo que esta técnica puede no ser la solución para resolver este problema.

Conclusiones

Tanto en el ejercicio 1 como en el 2 se puede observar como el aumento de accuracy en el algoritmo de Aprendizaje en Batch se incrementa más despacio que el de Secuencia. Esto se debe a que en el algoritmo de Aprendizaje en Batch los pesos se actualizan después de que el modelo ha revisado todos los datos y lo hacen tantas veces como muestras en el dataset, pero el arreglo final de los pesos se probará una sola vez para todas las muestras en la siguiente época; mientras que en el Algoritmo por Secuencia, los pesos se actualizan inmediatamente después de que se envía un input de datos al perceptrón, de manera que al final de una época los pesos se han actualizado y han sido probados una cantidad de veces igual al tamaño del dataset; es por ello que en el Algoritmo en Secuencia obtiene una precisión alta mucho más rápido que el Algoritmo en Batch.

Por otro lado, al momento de utilizar un modelo de ML es importante asegurar que la certeza con la que el modelo predice es mayor a la probabilidad de escoger una de las categorías al azar. Esto se pudo ver en el ejercicio 3, donde la probabilidad de que la línea recta encontrada pueda predecir la categoría a la que una nueva muestra pertenecería era un poco menor a la del 50%, por lo que da exactamente igual usar el algoritmo implementado, o solamente escoger un número al azar entre 1 y 2.