

INTELIGENCIA ARTIFICIAL II

TRABAJO PRÁCTICO N°3

INTEGRANTES:

- Aldao, Antonella 12670
- Berridy, Ignacio 11987

Ejercicio 2	2
Ejercicio 3	2
Ejercicio 4	3
Ejercicio 5	5
Ejercicio 6	9

Ejercicio 2

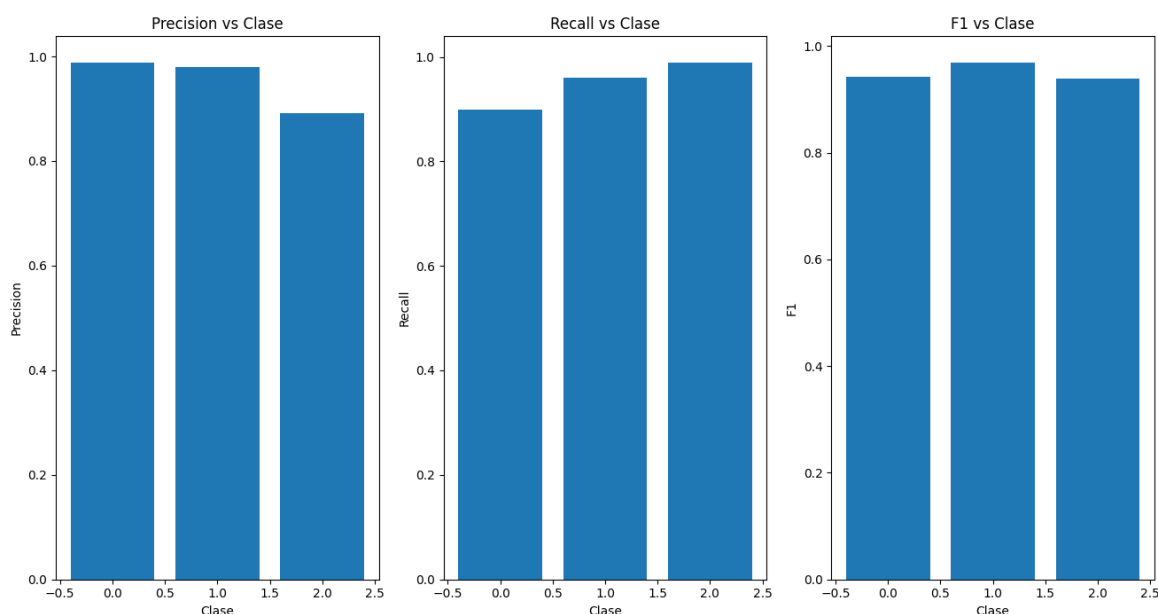
Se modificó la red neuronal para que la misma obtenga distintos valores que sirven como medida de precisión de la misma.

Para ello agregamos una función que calcula tanto la precisión, como el recall y el score-f1.

$$\text{Precisión} = \frac{\text{cantidad de ejemplos clasificados como } C \text{ correctamente}}{\text{cantidad total de ejemplos clasificados como } C}$$

$$\text{recall} = \frac{\text{cantidad de ejemplos clasificados como } C \text{ correctamente}}{\text{cantidad total de ejemplos de la clase } C}$$

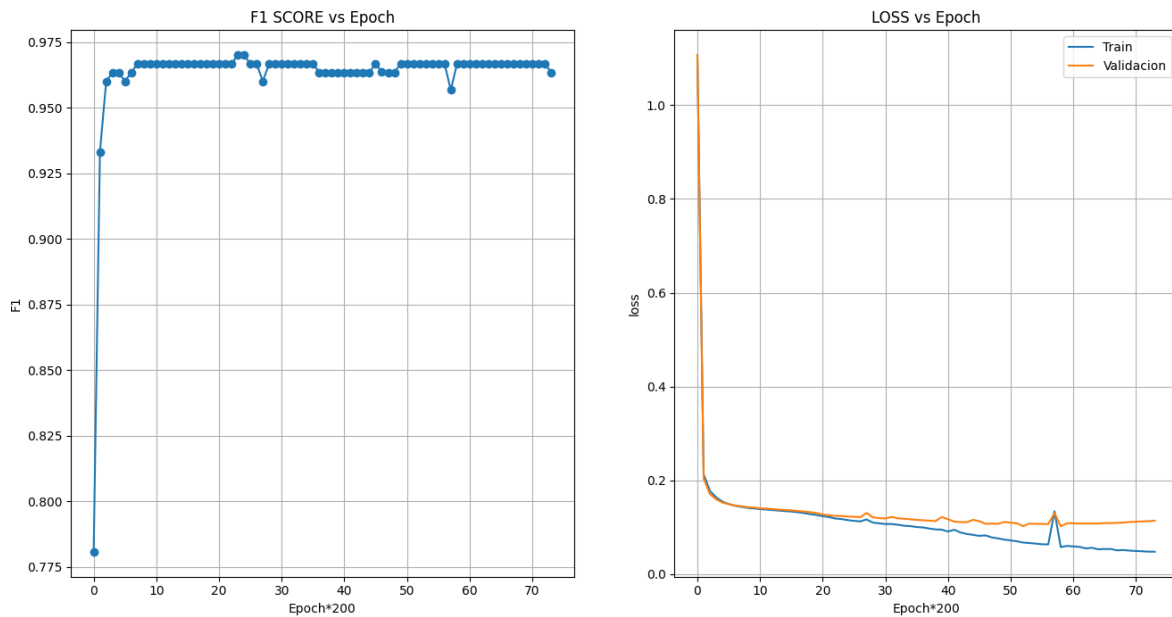
$$F1 = \frac{2 * \text{Precisión} * \text{recall}}{\text{Precisión} + \text{recall}}$$



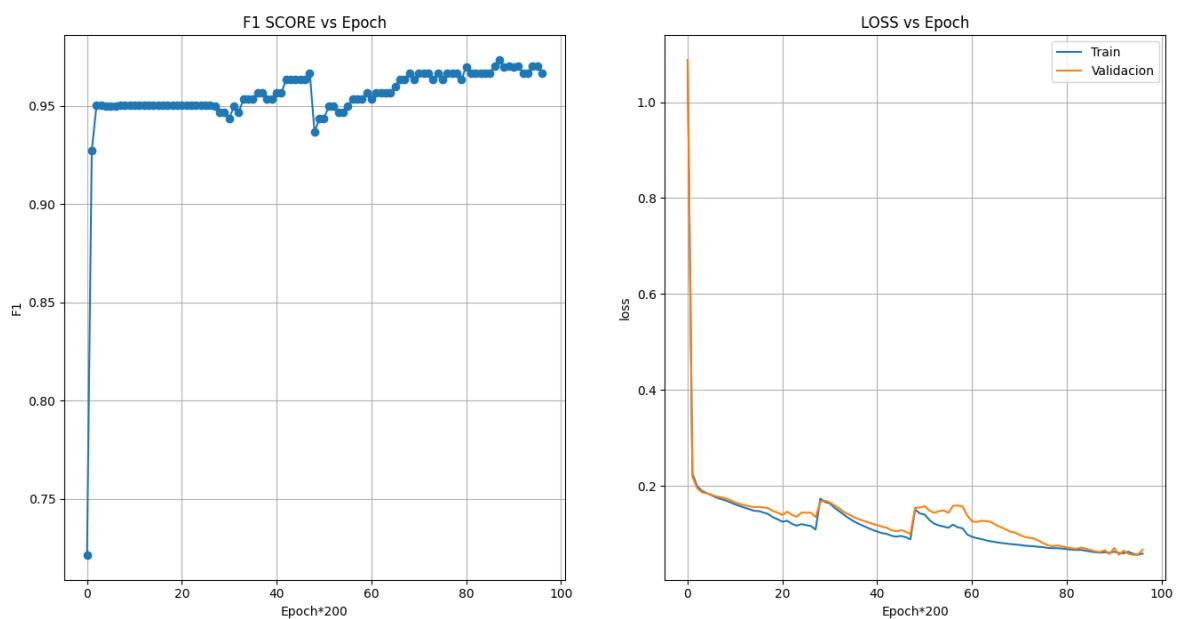
Ejercicio 3

Se implementó un criterio de parada temprana utilizando un conjunto de validación, distinto del conjunto de test y del de training.

Para el mismo se calcula la derivada numérica del score-f1 y de la función loss del conjunto de validación cada cierta cantidad de epochs, cuando la derivada del score-f1 si las últimas n derivadas de la función de loss son positivas, se detiene el entrenamiento.



En este primer gráfico vemos el criterio de parada implementado con las últimas 5 derivadas de la función de loss.

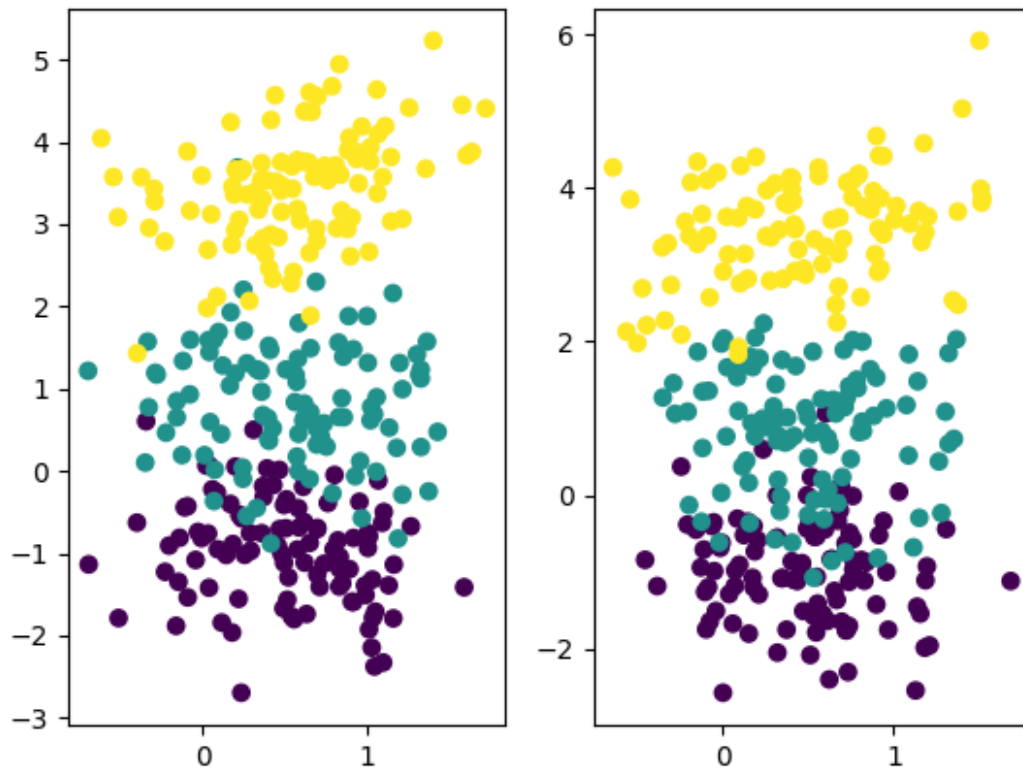


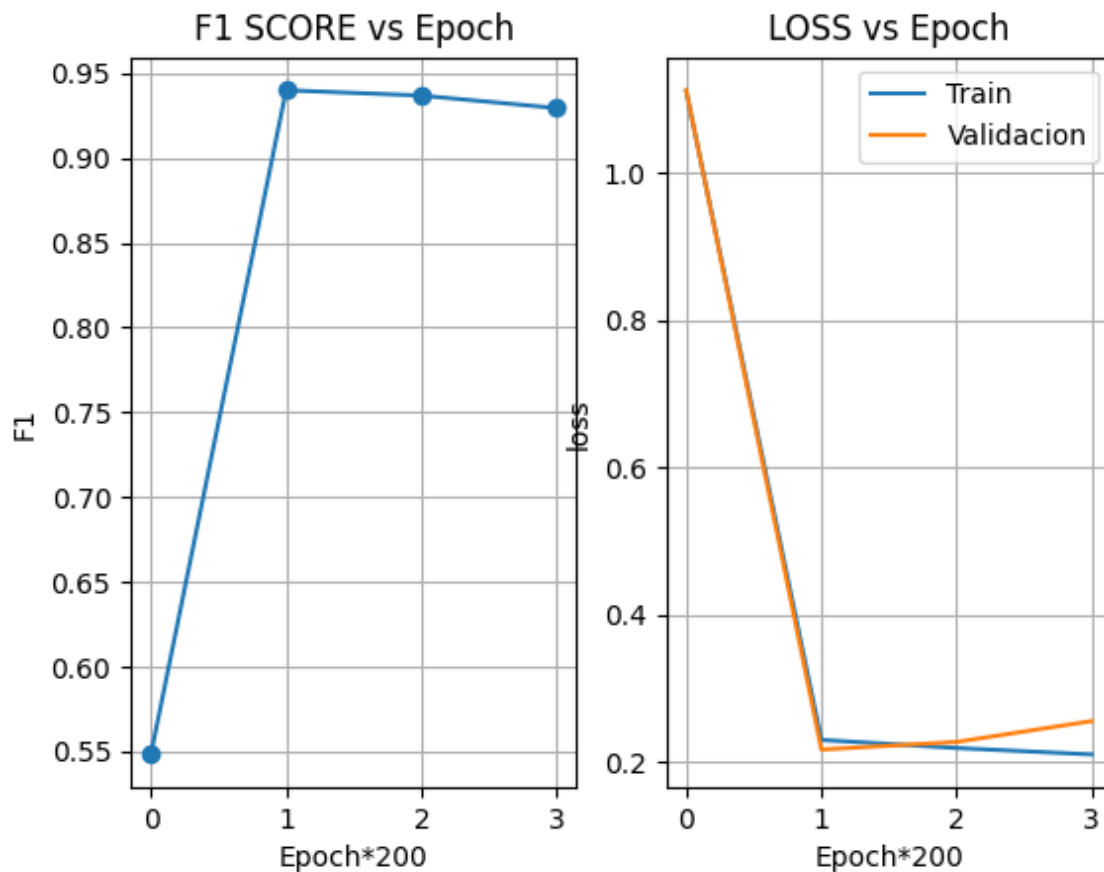
En este segundo gráfico vemos el criterio de parada implementado con las últimas 3 derivadas de la función de loss.

Ejercicio 4

Para este ejercicio se creó un nuevo generador de datos para poder ver como la red neuronal funcionaba con este cambio. A continuación se puede ver la nueva forma en la que se generan los datos. A la izquierda está el conjunto de entrenamiento y a la dera el

conjunto de validación, el cual se usa para la parada temprana explicada en el ejercicio anterior.





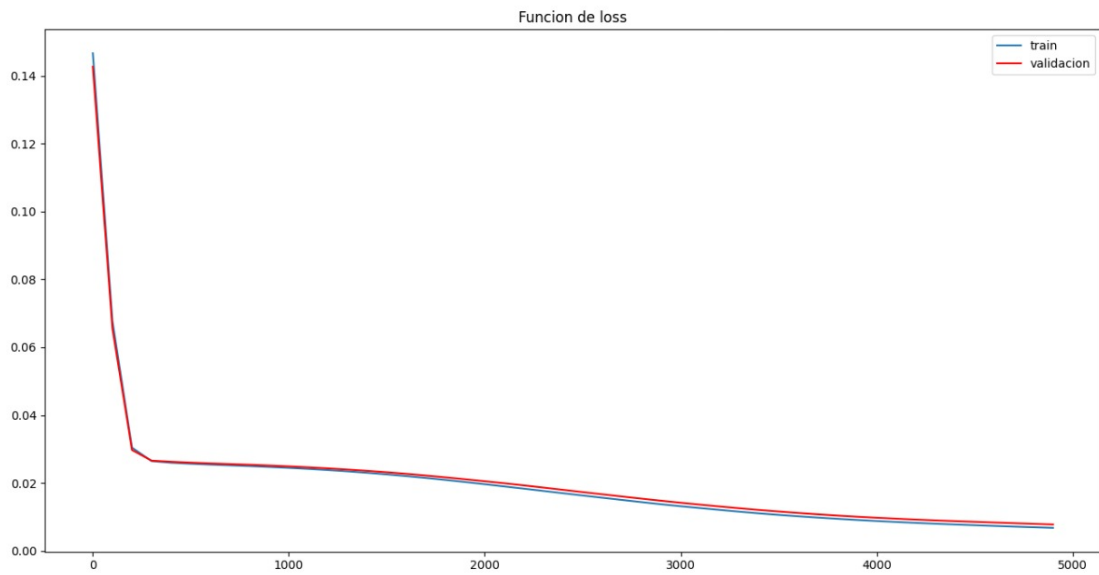
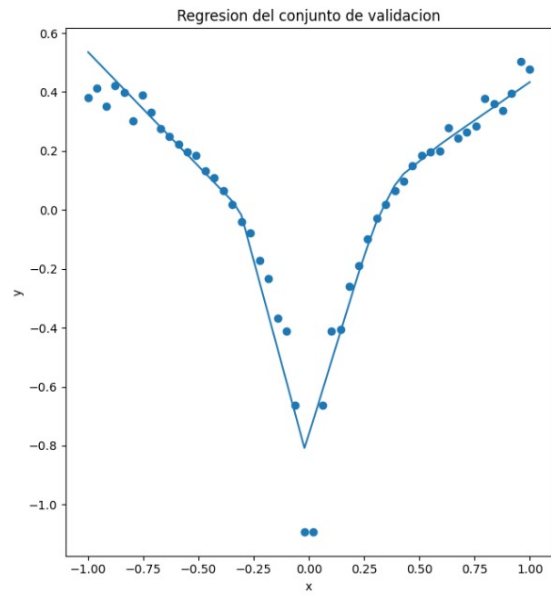
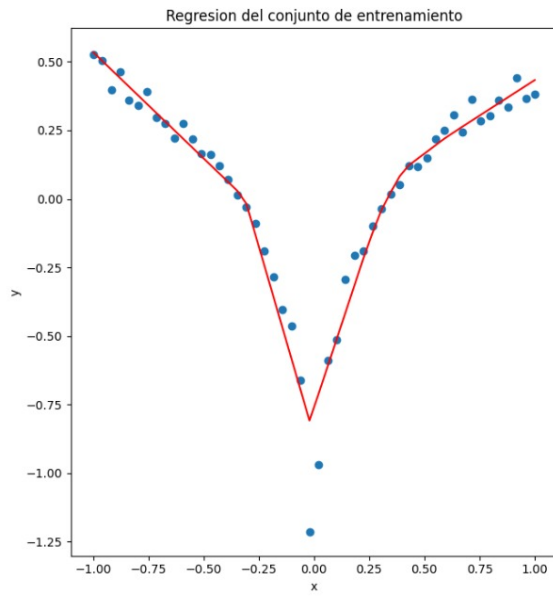
Ejercicio 5

Se modificó la red neuronal para que resolviera problemas de regresión. Para ello se utilizó una función de loss mse (mean square error), y su respectiva derivada para el método de los gradientes.

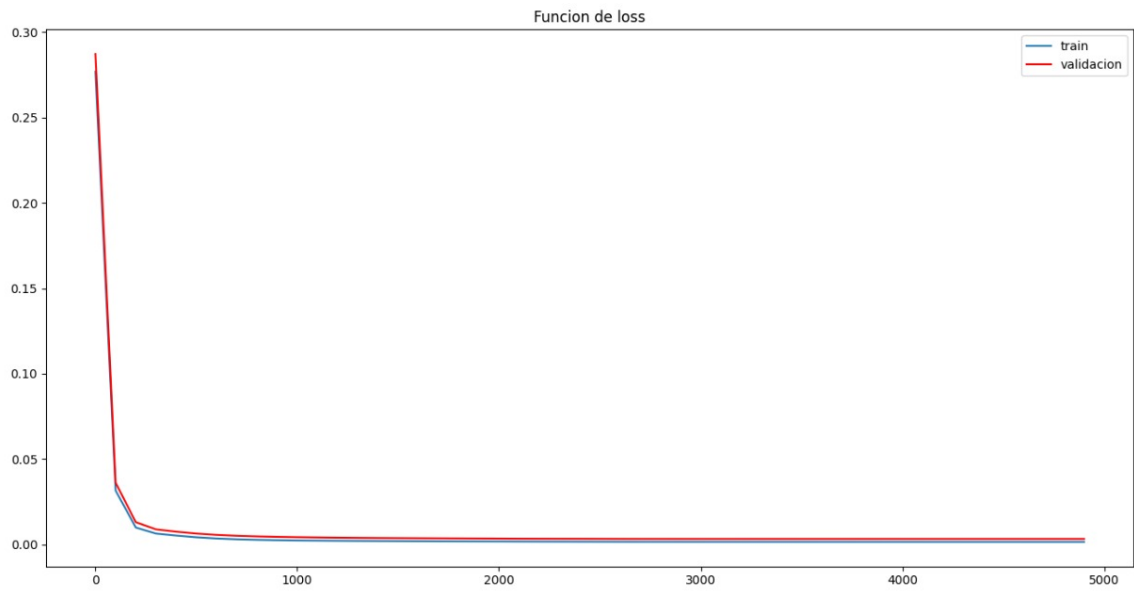
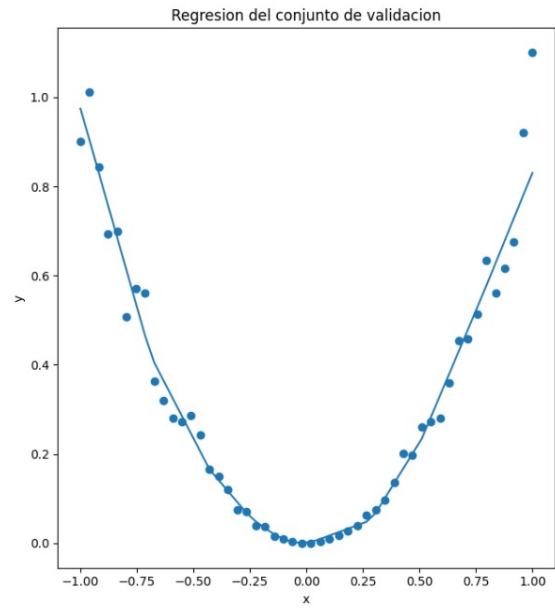
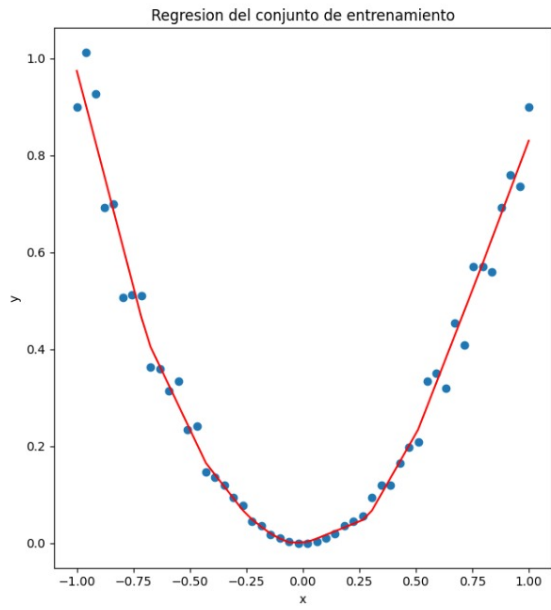
Además se cambió el generador de datos por uno que devuelve una nube de puntos alrededor de una función conocida, siendo cada punto de la nube entre el 80% y 120% del valor correspondiente a la función, dicho porcentaje varía de manera aleatoria.

Algunos ejemplos de la red para distintas funciones:

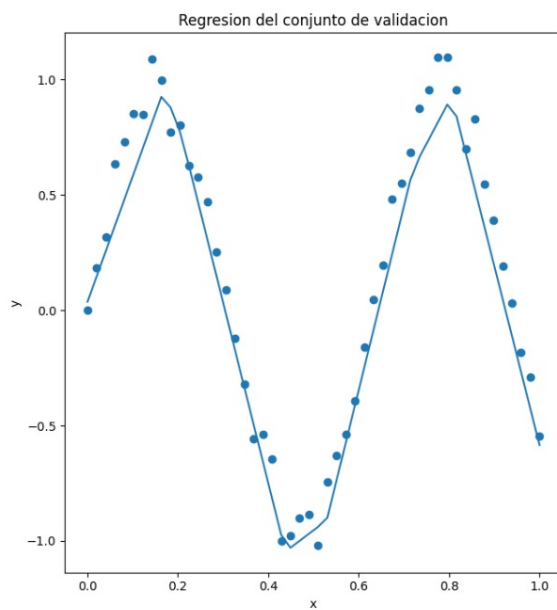
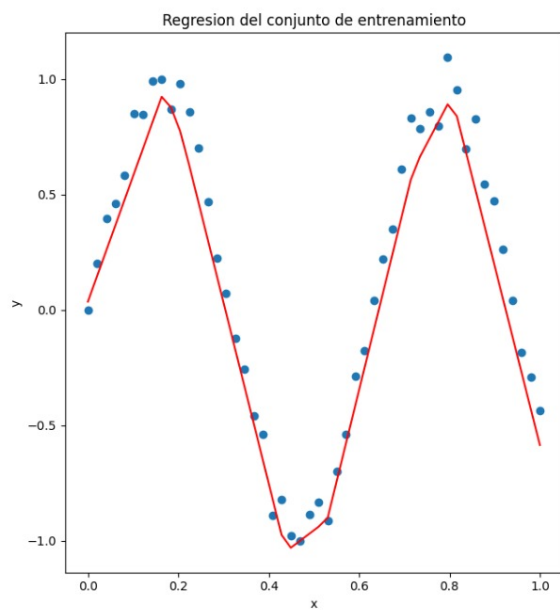
$$f(x) = \log_{10}(3x)$$



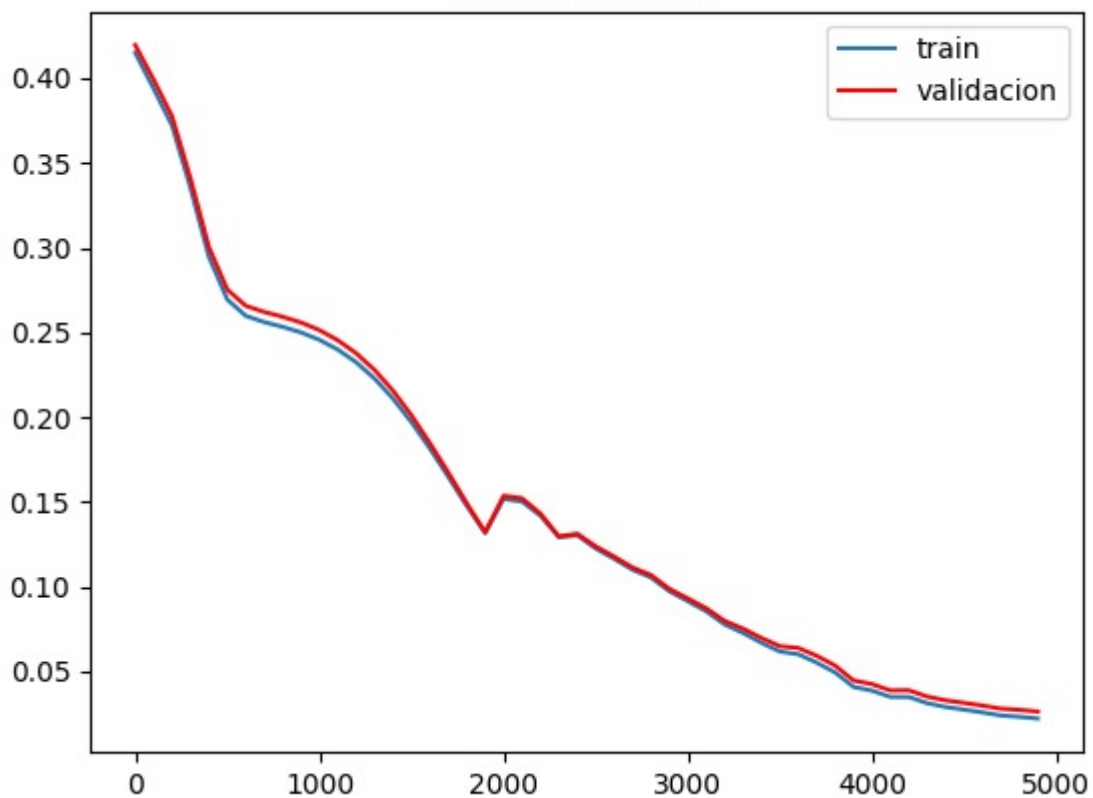
$$f(x) = x^2 + 2$$



$$f(x) = \sin(10x)$$



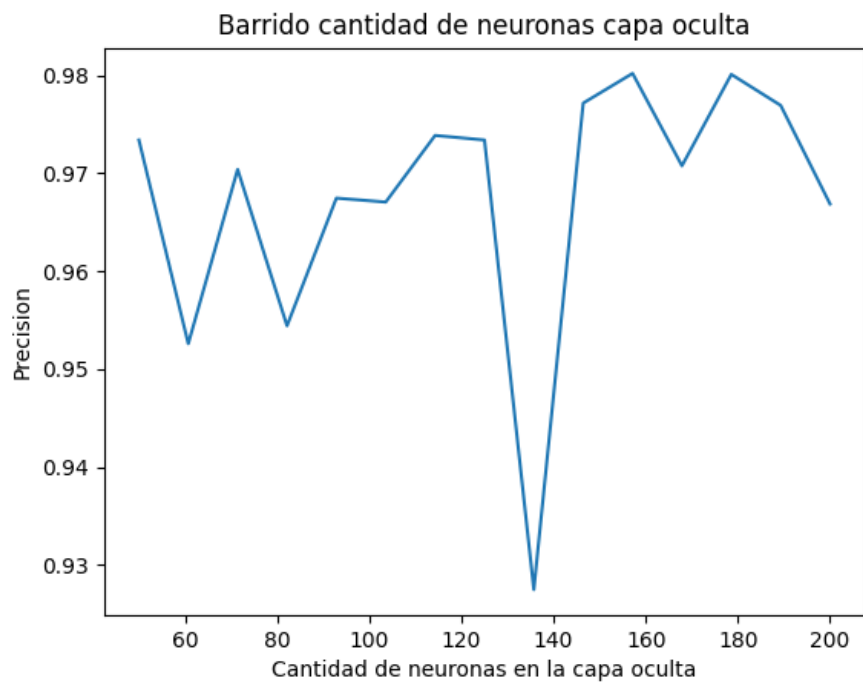
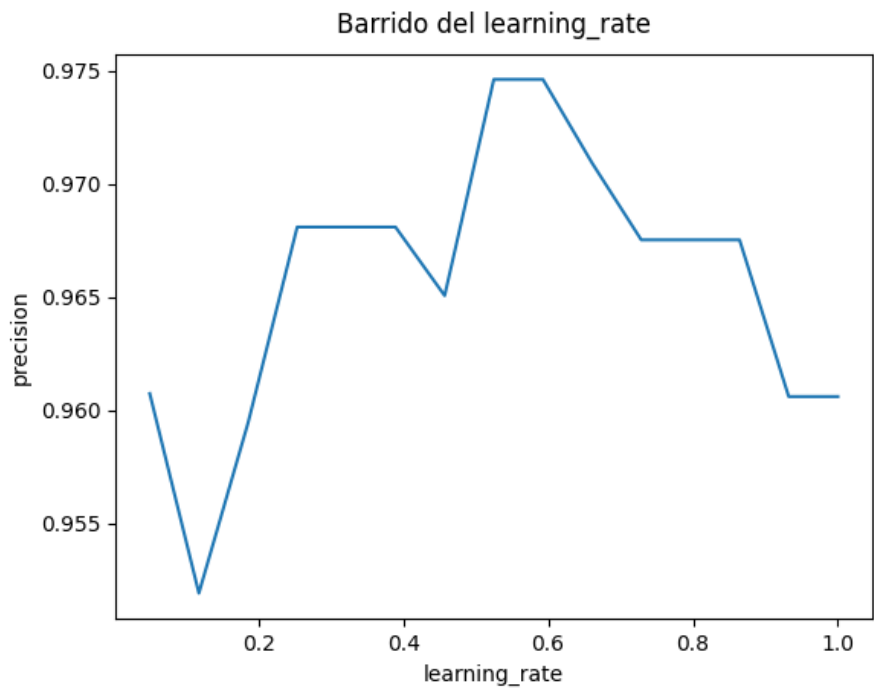
Funcion de loss



Ejercicio 6

Se realizó un barrido de parámetros para la cantidad de neuronas de la capa oculta y para el valor del learning rate, se analizó dichos parámetros en relación con la precisión obtenida para el conjunto de validación, luego del entrenamiento.

Se obtuvieron los siguientes resultados:



También se realizó un cambio de la función de loss por la función sigmoide.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Figure 1

