

Redes Neuronales

Práctica 5 - Aprendizaje no supervisado

F. M. Cabrera

13 de mayo de 2021

Ejercicio 1

Para este ejercicio se busco resolver el problema de XOR a partir de dos arquitecturas distintas, las cuales pueden observarse en la Figura ?? y cuya principal diferencia es que una presenta una reinyección de la entrada mientras que otra no. En ambos casos, la función de activación de la capa oculta fue tanh y como función de costo se utilizo MSE. Debido a que la función de activación utilizada nunca retorna 1 o -1 , se determino un umbral de 0.9 en la función de costo, de manera que si el resultado supera dicho umbral, se considera la salida como 1. De manera análoga, se determino un segundo umbral de -0.9 para determinar cuando la salida se considera un -1.

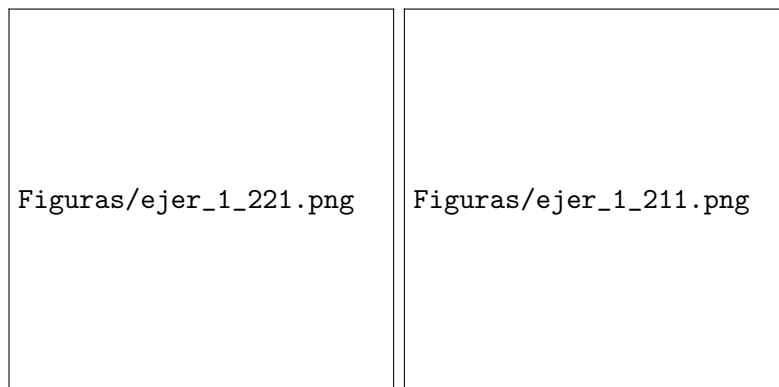


Figura 1: Esquema de las arquitecturas utilizadas para resolver el problema del XOR.

En ambos casos se utilizo la totalidad de los datos como datos de entrenamiento. Los resultados para la función de costo y el *accuracy* para la primer arquitectura en 10 procesos de entrenamiento independientes de la red pueden observarse en la Fig. ??, mientras que en la Fig. ?? se observan los resultados análogos para la segunda arquitectura. En ambos casos se observa que algunos procesos de entrenamiento no convergen y no alcanzan un valor de *accuracy* optimo del 100 %.

Se definió el tiempo de convergencia como aquella época en donde la función de costo alcanza un valor de 0,1, pero solo para los modelos que convergieron a una solución optima dentro de las 2000 épocas. Para comparar los tiempos de convergencia entre ambas arquitecturas, se entrenaron 1000 modelos independientes que lograron converger y se obtuvo su tiempo de convergencia, para ambas arquitecturas. Para la primer arquitectura se obtuvo un tiempo de convergencia promedio de 157,5 épocas, mientras que para la segunda se obtuvo un valor de 148,7. En principio, esta diferencia, aunque sea pequeña, podría indicar que la segunda arquitectura es mejor para la resolución del problema XOR. Sin embargo, para obtener 1000 modelos entrenados que hayan logrado converger a una solución optima, fue necesario entrenar 1288 modelos para la primer arquitectura, mientras que para la segunda fueron necesarios 1983. Es decir, que el 77,6 % de los modelos creados con la primer arquitectura lograron converger, mientras que para la segunda solo el 50,4 %, lo cual apunta a que la primer arquitectura es mas satisfactoria para resolver el problema.

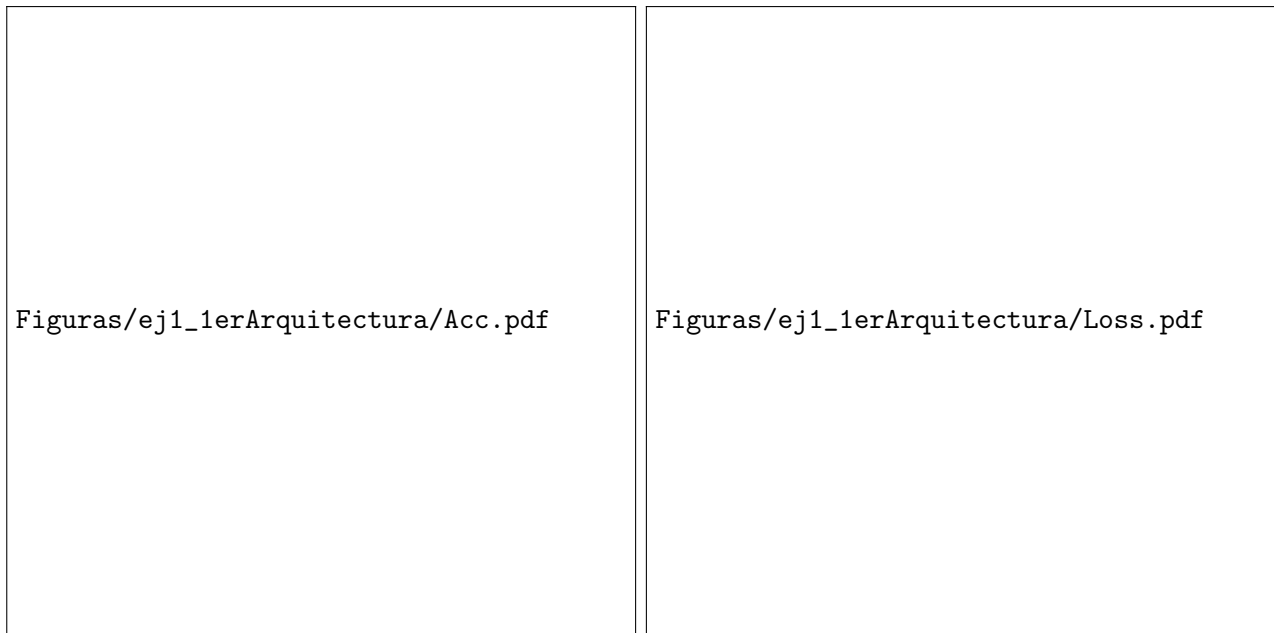


Figura 2: Función de costo MSE y *accuracy* para el entrenamiento de 10 modelos independientes, utilizando la primer arquitectura propuesta en Fig. ??.

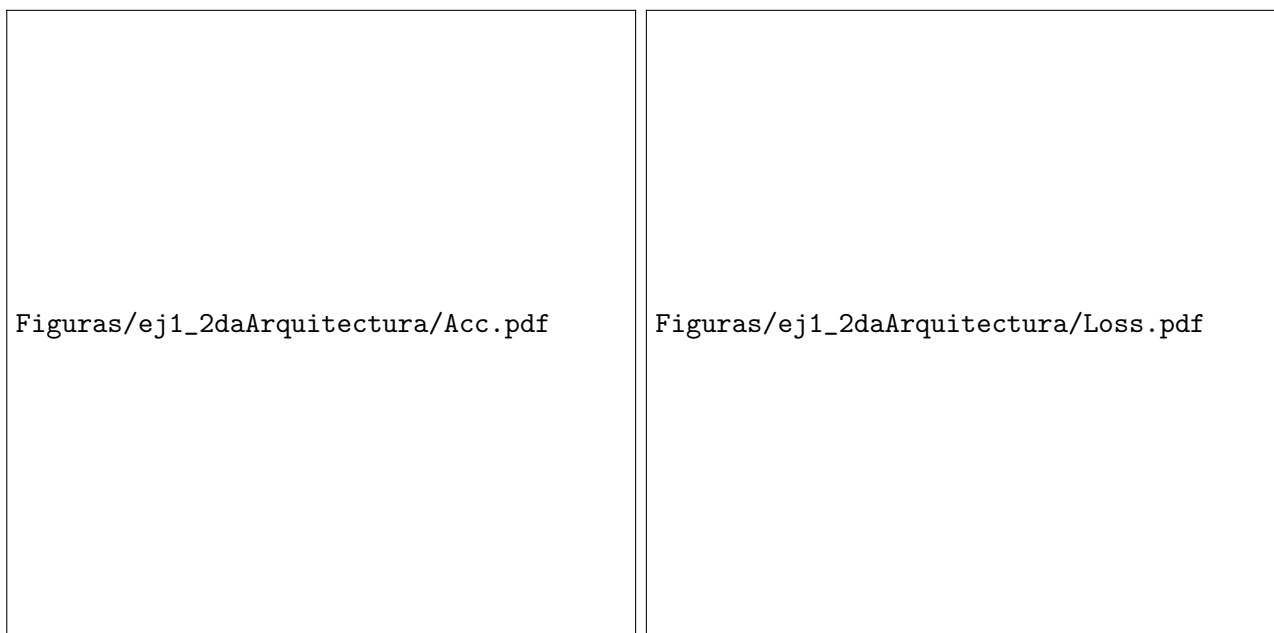
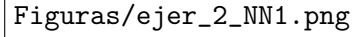


Figura 3: Función de costo MSE y *accuracy* para el entrenamiento de 10 modelos independientes, utilizando la segunda arquitectura propuesta en Fig. ??.

Ejercicio 2

Se busco estudiar una generalización del problemas de XOR para N entradas, en donde la salida esperada es el producto de todas las componentes de la entrada. La arquitectura utilizada consistió en una capa oculta de N' neuronas con una función de activación **tanh**, seguida de una capa de salida de 1 neurona tambien con activacion **tanh**. Debido a que la cantidad de datos de entrada es igual a 2^N y era de interés estudiar los casos en donde $N > N'$ y $N < N'$, se fijo N a un valor de 5 y se entreno la red para distintos valores de N' : 1, 3, 5, 7, 9 y 11, en todos ellos durante 5000 épocas.

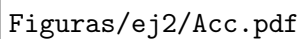
En la Figura ?? se observan los resultados obtenidos para la función de costo y *accuracy* en cada uno de los casos. En caso de que $N' > N$, los resultados son similares a los del ejercicio anterior,



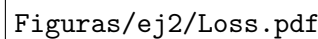
Figuras/ejer_2_NN1.png

Figura 4: Esquema de la arquitectura utilizada para resolver el problema del XOR generalizado.

ya que en una pequeña cantidad de épocas se obtiene un 100 % de precisión. La situación se vuelve mas problemática al disminuir N' , en donde se observa que para $N' = N = 5$ el aprendizaje de la red nunca alcanza una precisión total e incluso no mejora de manera monótona conforme avanzan las épocas. Los resultados son incluso peores cuando $N' < N$, en donde se requieren muchas mas épocas para que la red aprenda e incluso para $N' = 1$ nunca se logra alcanzar un valor de *accuracy* no nulo. Este problema es esperable, ya que el valor de N' determina las dimensiones de las matrices de pesos de cada una de las capas. Al disminuir N' , menor sera la cantidad de componentes de estas matrices, con lo cual deberá ajustar una gran cantidad de datos de entrada (2^N) con menos grados de libertad, que, en caso de ser demasiado pocos, se presentara un problema de underfitting y la red no mejorara sus resultados.



Figuras/ej2/Acc.pdf



Figuras/ej2/Loss.pdf

Figura 5: Función de costo MSE y *accuracy* para el entrenamiento de 6 modelos independientes variando la cantidad de neuronas de la capa oculta, utilizando la arquitectura propuesta en Fig. ??.