Informe Proyecto 1 Primera parte

03-35752 Gabriel Casique 06-40386 Jesus Torres 07-41120 Isaac Lopez
07-41092 Tony Lattke

8 de febrero de 2013

Para la realización de los experimentos se implementó en el lenguaje de programación python una serie de algoritmos de redes neuronales que tienen una sola neurona. Los experimentos llevados a cabo fueron:

Resumen

Las redes neurales constituyen una versátil herramienta en diferentes ámbitos de la computación moderna. Éstas permiten solucionar problemas de forma parecida a la que usamos los seres pensantes. Es decir aprendiendo y luego utilizando estas experiencias y datos para obtener un mejor desempeño en futuras ejecuciones.

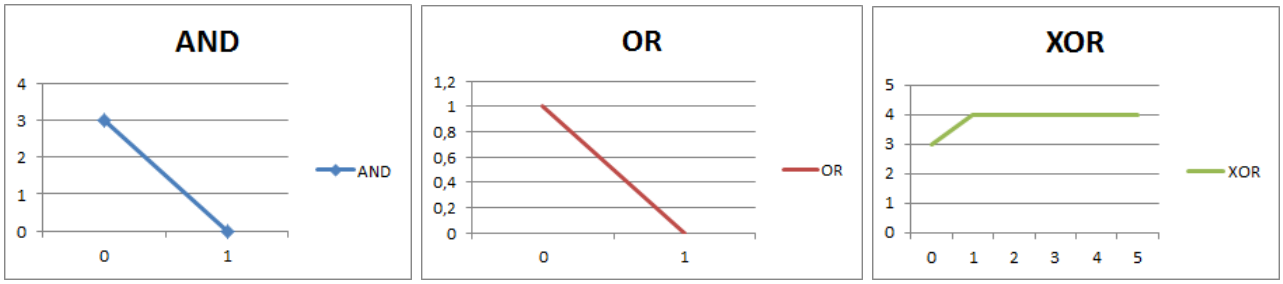
Para esta primera parte del proyecto se nos pide que implementemos algunos algoritmos de redes neurales que solo involucren una neurona. La idea es familiarizarse con las terminologías y los funcionamientos de las mismas y a su vez hacer un análisis que permita conocer las diferencias entre implementaciones y variantes que influyen directamente en el desempeño de los algoritmos.

Detalles de implementación/experimentación

Experimento 1

Implemente un perceptrón con n entradas. Entrénelo para que aprenda las siguientes funciones booleanas (usando 0;1 como tasa de aprendizaje): AND, OR, XOR. Grafique el error E como función del número de iteraciones de entrenamiento para cada una de las funciones. ¿Qué conclusiones puede obtener?

Respuesta:

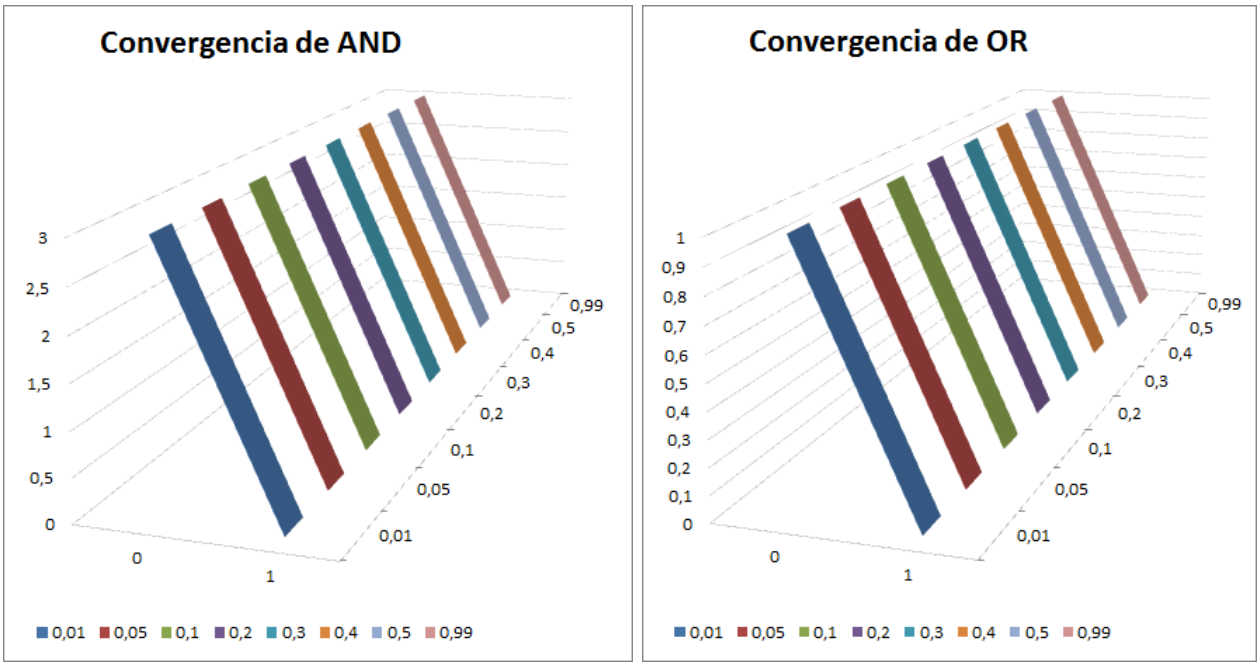


Se puede apreciar como para las primeras dos funciones (And y Or) el algoritmo es capaz de eliminar el error en una sola iteración. Sin embargo para el Xor, no es posible disminuir el error. Ésto ocurre debido a que la función Xor no es linealmente separable y el perceptron no puede determinar los pesos adecuados.

Experimento 1, parte a

a) Para las funciones AND y OR pruebe los siguientes valores para la tasa de aprendizaje (η): 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 y 0.99. ¿Qué conclusiones puede obtener?

Respuesta:

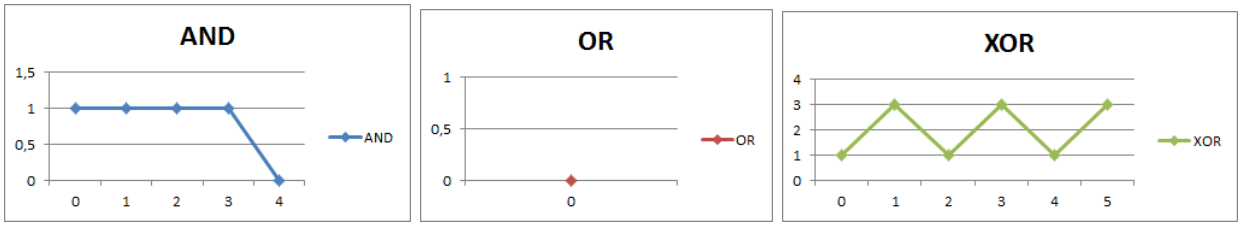


Viendo los resultados del algoritmo se puede afirmar que sin importar la tasa de aprendizaje, el perceptron fue capaz de conseguir los valores esperados en solo una iteración. Por lo tanto para estas funciones indiferentemente de la tasa de aprendizaje este algoritmo resuelve en los mismos tiempos.

Experimento 2

Implemente la regla de entrenamiento delta para una neurona artificial de n entradas (unidad lineal). Entrénela para que aprenda las siguientes funciones booleanas: AND, OR, XOR. Grafique el error E como función del número de iteraciones de entrenamiento. ¿Qué conclusiones puede obtener?

Respuesta:



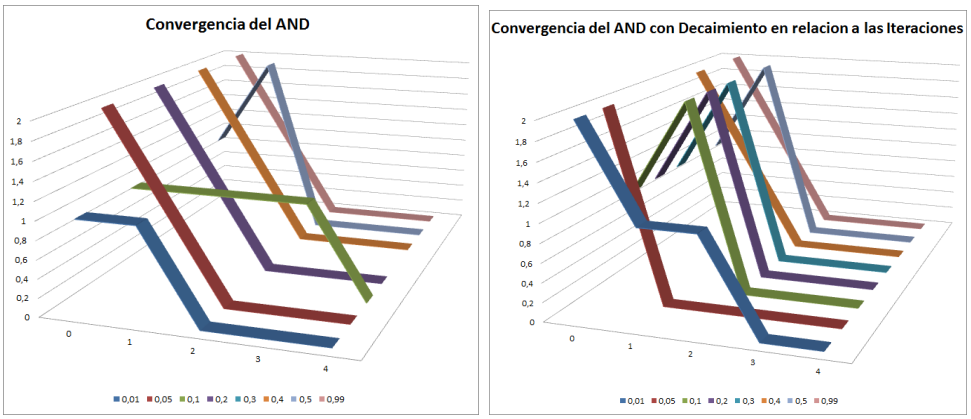
En el caso de las funciones booleanas AND y OR el error es superado en pocas iteraciones de la misma. Para la función XOR el error se mantiene oscilando entre los valores 1 y 3 en todas las iteraciones probadas por lo que se concluye que no converge indiferentemente sea la tasa evaluada, dado que no es linealmente separable.

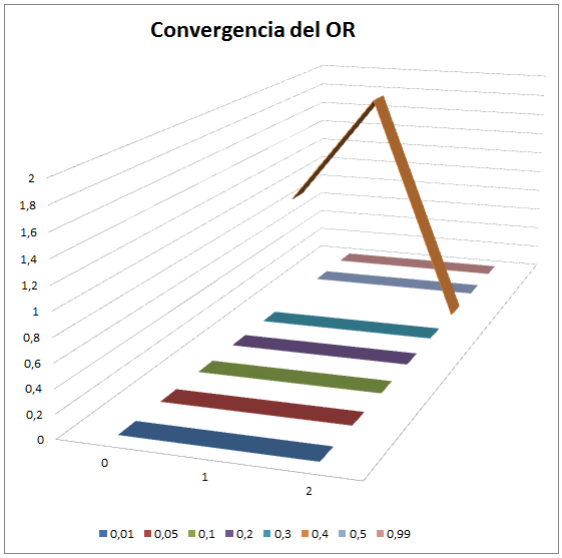
Experimento 2, parte a

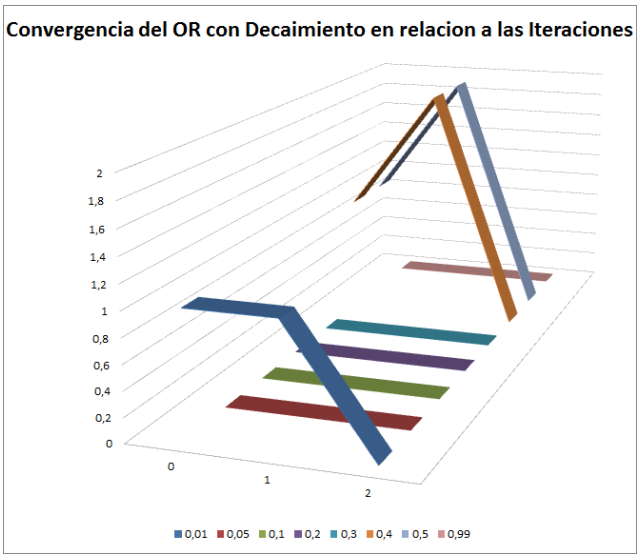
Pruebe los siguientes valores constantes para la tasa de aprendizaje (η): 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 y 0.99. Compare los resultados cuando se utiliza una tasa de aprendizaje que decaiga como η/i para la i-ésima iteración. ¿Qué conclusiones puede obtener?

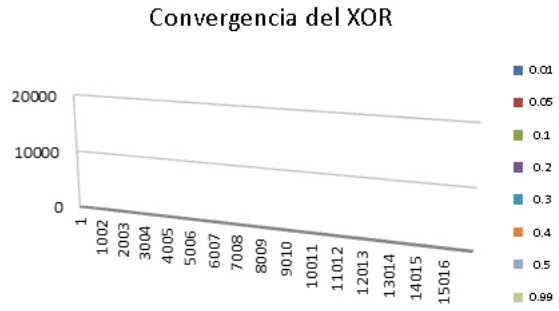
Respuesta:

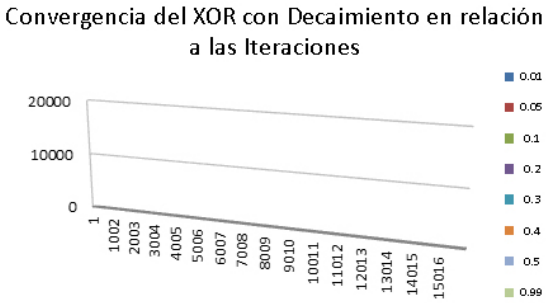
Para este algoritmo se puede apreciar diferencia entre la velocidad de aprendizaje para las distintas funciones y los distintos valores evaluados. En los casos de las funciones AND y OR, éstas convergen en considerablemente pocas iteraciones. La Función AND converge para todas las tasas en un máximo de 4 iteraciones, mientras para la función OR converge para la primera iteración o la segunda. Ahora bien, para el caso de la función booleana XOR no se consigue convergencia.







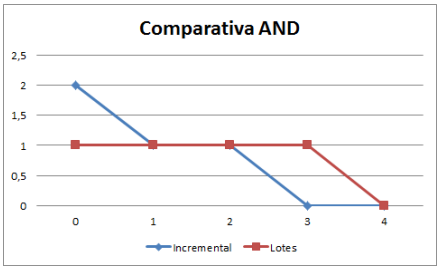


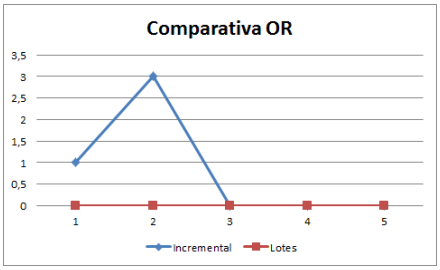


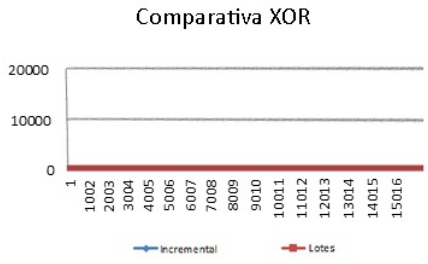
Experimento 2, parte b

Pruebe con aprendizaje incremental y por lotes. ¿Cuál converge más rápido? Considere el número de actualizaciones de pesos y el tiempo total de ejecución.

Respuesta:







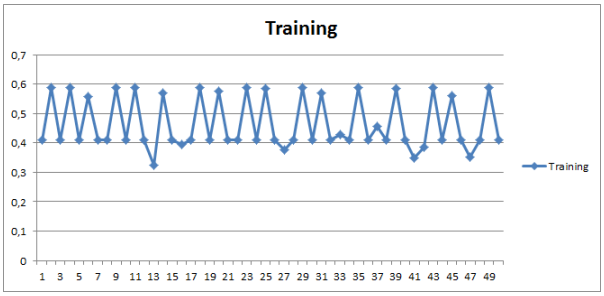
Gracias a las gráficas podemos observar que el aprendizaje por lotes tiene menos errores y converge más rápido, inclusive en el caso del OR converge en la primera iteración.

Experimento 3

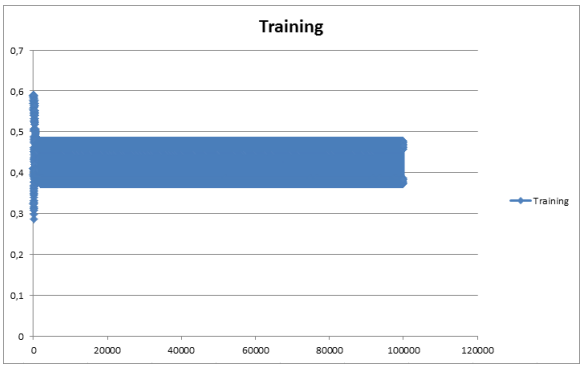
Entrene su neurona Adaline (unidad lineal de regla delta) para que clasifique los datos del conjunto Liver Disorders Data Set (http://archive.ics.uci.edu/ml/datasets/Liver+Disorders). Separe este conjunto en entrenamiento y prueba. Grafique ambos errores. ¿Cuál es la mejor tasa de error (entrenamiento y prueba) que pudo obtener?.

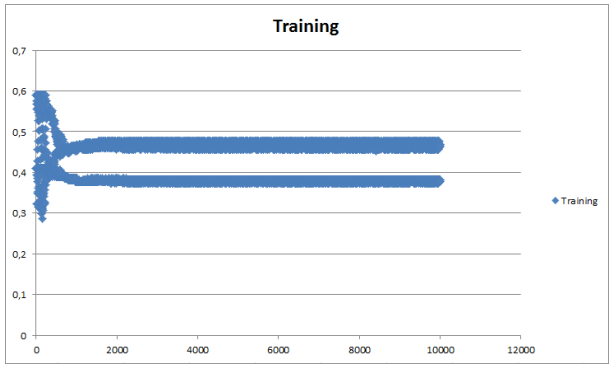
Respuesta:

Para realizar este experimento se tomó el 70 % de las muestras aleatoriamente para un conjunto de entrenamineto y el resto se dejo para el conjunto de prueba. Graficando hasta 50 iteraciones se puede ver que el error esta acortado por 0,6 y 0,3.

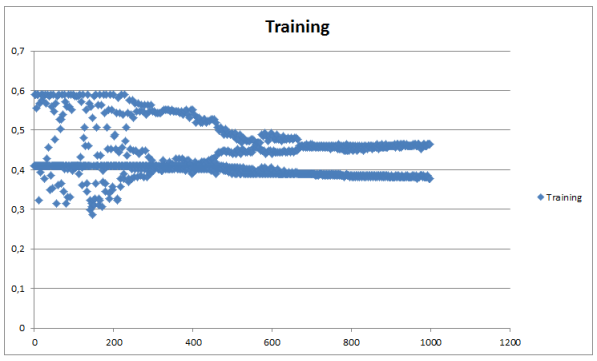


Los graficos siguientes muestran la de dispercion del error:

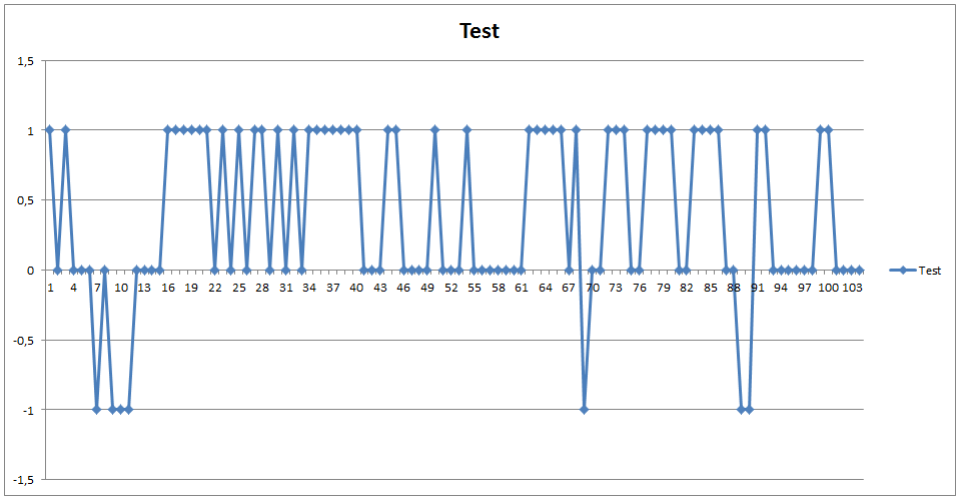




Si graficamos todos los puntos de los 10000 iteraciones que hicimos tenemos los resultados de la primera imagen, en la que se puede ver que luego de cierto punto el error se acorta mucho mas y todos los puntos se ve en una sola franja, este punto es el que deberiamos buscar como el óptimo de este modelo. Pero al realizar un acercamiento en las 10,000 iteraciones, aun podemos hacer mas zoom obtenemos la segunda imagen.



Haciendo un acercamiento en las mil iteraciones vemos que en este punto es que la neurona consigue realmente su mejor error posible el cual esta acotado claramente entre 0,5 y 0,35. Por lo tanto esto nos da a pensar que la mejor máquina estaría aproximadamente en la iteración 700.



Ya comparando con nuestro grupo de prueba el error encontrado fue al rededor del 50 % y 60 % para todas las corridas que hicimos, por tanto podemos decir que el nuestra red neural acerto solo la mitad de las veces que se probó, esto puede deberse a que soló se esta contando con una sola neurona.

Conclusiones

Podemos concluir que el uso de una sola neurona en para todos los experimentos logró estimar resultados muy bien para el caso de estudio de operaciones lógicas, más no lo se comportó de la misma forma para el caso de estudió de Adaline y esto se puede deber a que teniamos una sola neurona operando los datos o que el conjunto de entrenamiento seleccionado no fue el mejor, aunque este fue seleccionado aleatoriamente.

Otro punto a destacar es que el variar el valor de la taza de aprendizaje para los distintos experimentos logró mostrar que el incremento de su valor lo que hace es incrementar el error, así que es aconsejable trabajar con la taza de aprendizaje más pequeña posible y así se encontran resultados más rápido ya que se converge a la solución en menor cantidad de iteraciones.