



Sistemas de inteligencia artificial

(1er cuatrimestre 2018)

Trabajo Práctico 1

Redes Neuronales

Julián Benítez 56283

Nicolás Marcantonio 56288

Eliseo Parodi Almaraz 56399

Índice

Índice	1
Introducción	2
Implementación	2
Detalle del algoritmo	2
Resultados	3
Elección de parámetros de la red	5
Conclusiones	6
Análisis de resultados	6
Arquitectura óptima	7

Introducción

En este informe se describe la implementación de una red neuronal multicapa cuyo aprendizaje es supervisado. A esta red se le enseña a partir de una latitud y longitud la altura de un terreno del cual debe llegar a una aproximación. Para aprender estos terrenos se hicieron diferentes pruebas de arquitectura y se varían los parámetros de la red y de las mejoras implementadas.

Implementación

Se realizó un programa en Octave, que a partir de un archivo recibe la configuración de los parámetros de la red neuronal.

La red neuronal se encuentra en un archivo llamado *Perceptron.m* que contiene la clase *Perceptron*. Esta clase representa la red neuronal. Esta red está representada por una colección de matrices dentro de la clase. Además el perceptrón contiene la constante de proporcionalidad del aprendizaje (η), la función de activación con su derivada y otros parámetros más utilizados para las mejoras aplicadas a los algoritmos de aprendizaje. Las mejoras que se aplicaron a los algoritmos de aprendizaje son las de momentum y η adaptativo.

Este perceptrón puede aprender los datos de manera incremental o con batch, llamando a los métodos *learnIncremental* y *learnBatch* respectivamente, o sus versiones con tasa de η adaptativo, *learnIncrementalAdaptative* y *learnBatchAdaptative*.

Detalle del algoritmo

El algoritmo comienza inicializando los pesos de las conexiones entre neuronas con números *aleatorios* entre -0.5 y 0.5. Luego, entra en un ciclo, el cual tiene una longitud indeterminada hasta dar con su condición de corte, la cual es que el error cuadrático medio sea menor a una cota especificada en el archivo de configuración. En cada uno de estos ciclos, se itera por un porcentaje de los patrones de entrenamiento, el cuál también es configurable. Los patrones son ordenados de forma aleatoria al cargar los datos, para no siempre entrenar con los mismos patrones.

Al terminar el entrenamiento de un conjunto de patrones, se procede a calcular el error cuadrático medio para estos patrones, y en caso de que sea menor a la cota, sale del ciclo. Finalmente al salir del ciclo prueba contra los patrones que no entrenó para ver el porcentaje de error en lo que aprendió la red.

Resultados

Se probó con la mejora del momentum para nuestra red neuronal con y sin método adaptativo para el método de aprendizaje incremental pero la red caía en un mínimo local, por lo que sus resultados no fueron tomados en cuenta, aunque se sugiere agregar ruido a la red como mejora para evitar caer en estos mínimos. Este trabajo no estudia ese tipo de mejoras.

Se compararon los resultados para el método incremental y el *batch* para el método con y sin la mejora del método adaptativo. Para estas pruebas se utilizó una red neuronal con una capa oculta de 40 neuronas y los siguientes parámetros, $\eta = 0.09$, la función de activación $g(x) = \tanh(x)$, $\beta = 1$, ϵ en el testeo = 0.05 y 90% de los patrones entrenamiento y 10% de prueba. Para el método adaptativo se usó un incremento de 0.05 y un decremento de 0.1.

Disminución del error según la época (incremental)

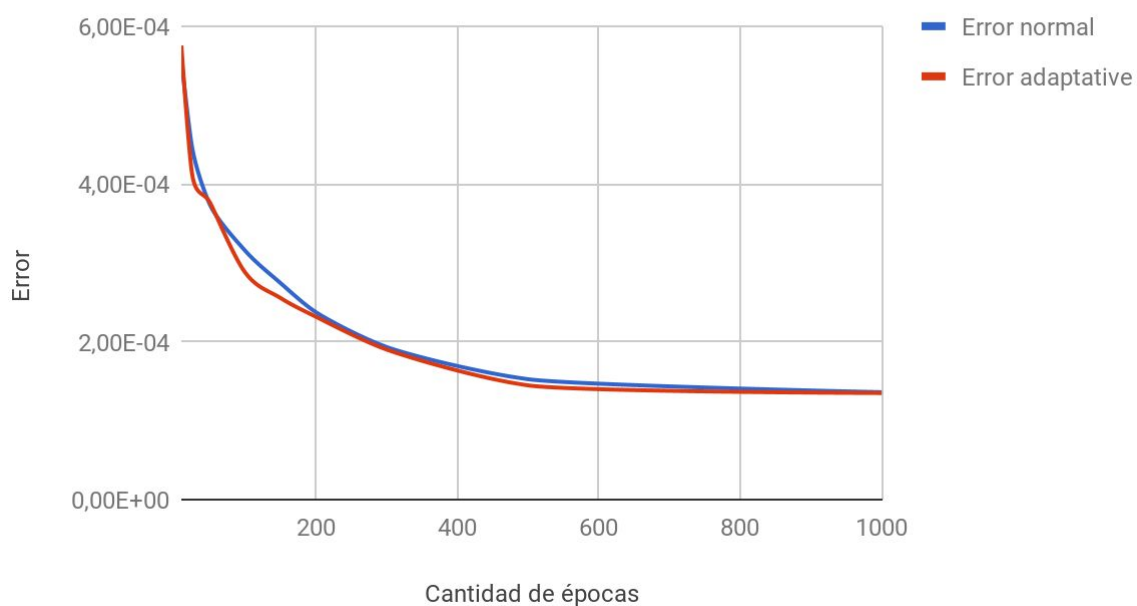


Figura 1: Error en función de la cantidad de épocas para el método incremental.

En este gráfico podemos ver que al principio la curva de error decrece más rápido con el método adaptativo pero luego se estabiliza y la diferencia es mínima. Por lo tanto no hay mucha diferencia entre elegir un método y otro pero el adaptativo tiene un error un poco más bajo a medida que las épocas avanzan.

En el gráfico se ve que la diferencia entre Batch y Batch + adaptative es mínima, efecto que también observamos en el gráfico anterior. Lo que más llama la atención de el método batch es la gran cantidad de épocas necesarias para aprender (de dos órdenes de magnitud más grande) en comparación al método incremental. Este efecto no es tan perjudicial ya que cada época se calcula mucho más rápido que en el modo incremental. Al contrario que en el modo incremental, el momentum se pudo aplicar y obtuvimos buenos resultados usándolo. Como se ve en el siguiente gráfico, momentum causó una mejora en la velocidad de aprendizaje. Solo la mejora de momentum causa que no aprenda tan rápido como la combinación de mejoras, pero a largo plazo es tan buena opción como el resto. El adaptative ayuda a que aprenda mas rapido inicialmente que es una desventaja de la mejora de momentum.

Disminución del error según la época (batch)

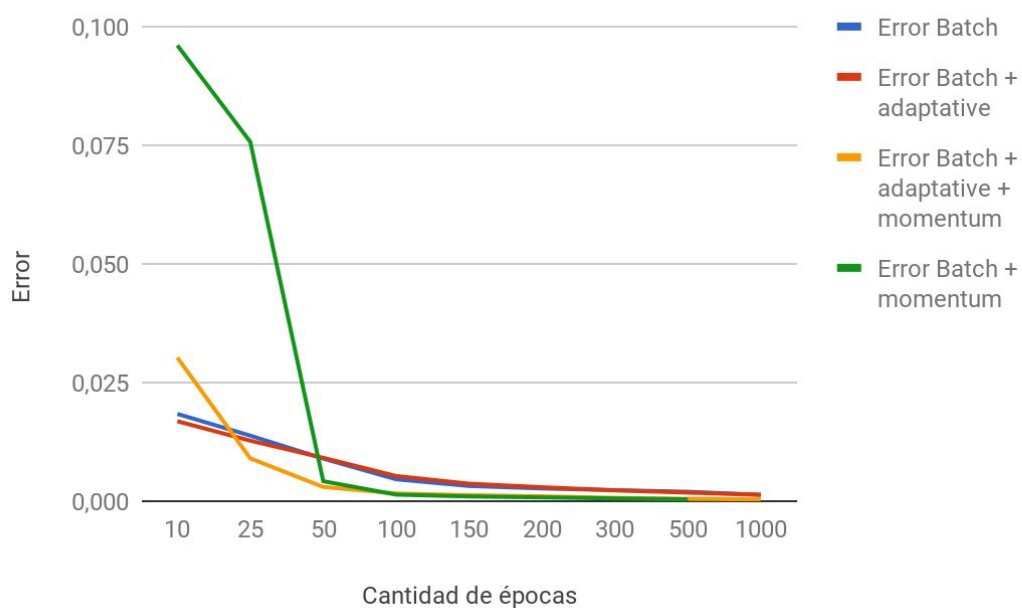


Figura 2: Error en función de la cantidad de épocas para el método batch.

Disminución del error según la época (incremental)

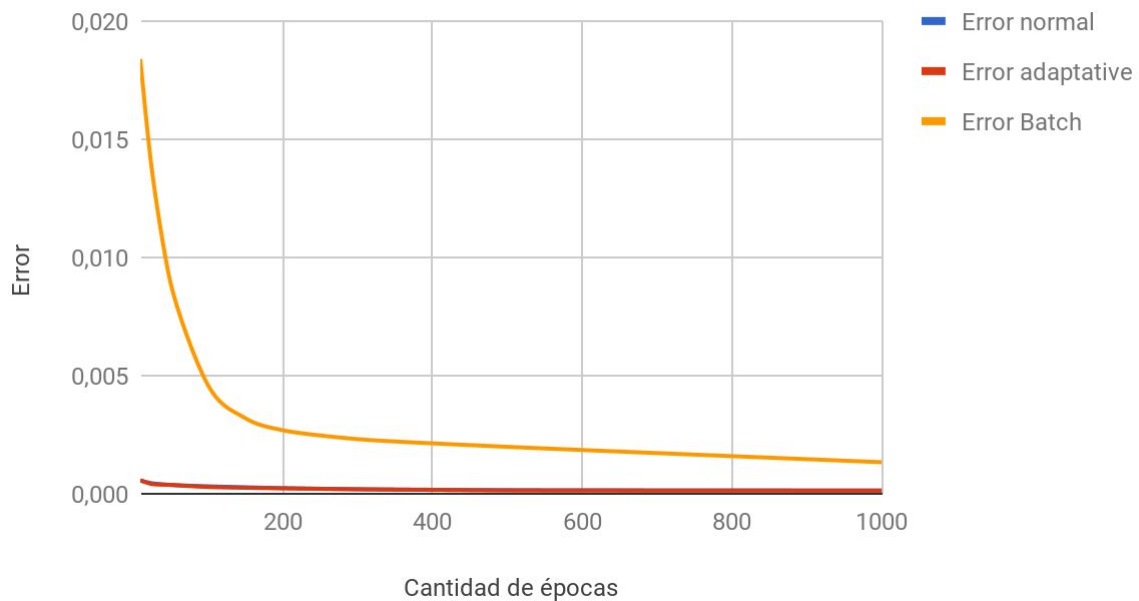


Figura 3: Error en función de la cantidad de épocas para método batch e incremental.

Comparando el caso del batch y del incremental se llegó a la conclusión que para nuestro terreno el método batch era ineficiente y no conllevaba ninguna mejora, así que se decidió utilizar el método incremental junto con el método de η adaptativo.

Elección de parámetros de la red

Se corrieron muchas pruebas variando el learning rate, el incremento y el decremento para ver que combinación de parámetros funcionaba mejor para nosotros. Analizando los datos obtenidos, se llegó a la conclusión que un Incremento de 0.05 y un decremento de 0.1 era lo óptimo para optimizar nuestras redes.

Épocas	Error de la Condición de corte	Learning Rate	Learning Rate Increment	Learning Rate decrement
8	1,00E-03	0,09	0,025	0,01
36	5,00E-04	0,09	0,025	0,01
9	5,00E-04	0,09	0,01	0,025
9	5,00E-04	0,09	0,01	0,01
7	5,00E-04	0,09	0,01	0,01
20	5,00E-04	0,09	0,009	0,018

67	5,00E-04	0,01	0,0001	0,0001
60	5,00E-04	0,01	0,0001	0,0001
71	5,00E-04	0,01	0,0001	0,01
89	5,00E-04	0,01	0,001	0,01
54	5,00E-04	0,01	0,01	0,01
54	5,00E-04	0,01	0,01	0,01
96	5,00E-04	0,01	0,01	0,01
12	5,00E-04	0,05	0,01	0,01
26	5,00E-04	0,05	0,01	0,025
737	2,50E-04	0,02	0,01	0,01
723	2,50E-04	0,02	0,01	0,1
441	2,50E-04	0,02	0,005	0,1
686	2,50E-04	0,02	0,05	0,1
221	2,50E-04	0,09	0,05	0,1
43	3,50E-04	0,09	0,05	0,1

Tabla 1: Resultados de la variación de parámetros para una red neuronal con método de aprendizaje incremental y adaptativo.

También se comparó dos arquitecturas distintas para ver cual es la óptima y se llegó a la conclusión que la arquitectura 1 (una capa de 40 neuronas) es la más adecuada para nuestro problema. La arquitectura 2 (dos capas de 12 neuronas), la segunda no estaba a la altura de la primera arquitectura.

Conclusiones

Análisis de resultados

A continuación se exponen los análisis de los resultados obtenidos:

- La red obtuvo una mejor performance con una sola capa oculta que con dos capas ocultas.
- La función de activación tangencial resultó mejor que la exponencial.
- El modo de aprendizaje batch no resultó de la manera que se esperaba, mostrando una clara deficiencia contra el incremental, si bien cada época era realizada mucho más rápido la relación época/error cuadrático medio nos hizo elegir el modo incremental.

- La optimización con momentum sobre la red presenta mejoras con respecto a la red sin optimizaciones pero su performance no es tan buena como con el η adaptativo.
- La combinación momentum η adaptativo no resultó mejor que la optimización del η adaptativo solo.
- Con tan solo 30-40 épocas se logró llegar a un resultado bastante razonable con más de un 90% de éxito.
- La principal deficiencia de la red se encuentra en los extremos del intervalo, esto puede verse claramente en la figura 4, la cual fue realizada con la arquitectura óptima.

Arquitectura óptima

Luego de realizar las comparaciones mostradas anteriormente, se llegó a la conclusión de que la arquitectura óptima para nuestro terreno en particular es la siguiente:

- η : 0.09
- g : tanh
- β : 1
- Parámetros Adaptativos -> incremento 0.05 decremento 0.1
- 1 capa oculta con 40 neuronas
- ECM de corte : 0.00035
- ϵ en el testeo : 0.05
- 90% de entrenamiento y 10% de prueba

Los resultados de la red fueron los siguiente, el aprendizaje terminó al cabo de 43 épocas con un porcentaje de éxito de 93.33%.

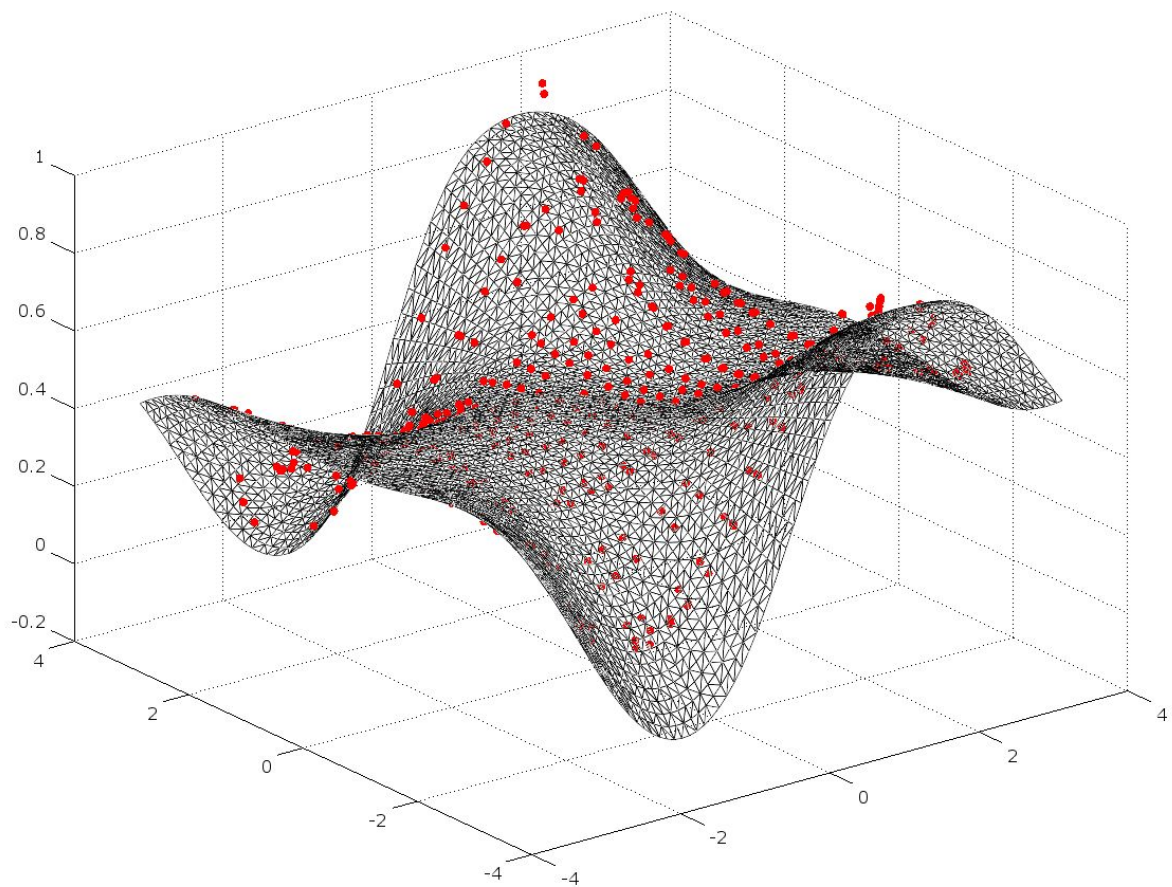


Figura 4: Terreno real (puntos rojos) sobre el terreno generado por la red neuronal.

Se puede observar en este gráfico que en las puntas el aprendizaje de la red es más difícil debido a que la red suaviza el terreno. A pesar de estas discrepancias con el terreno original que son inevitables para nuestro modelo de red neuronal.

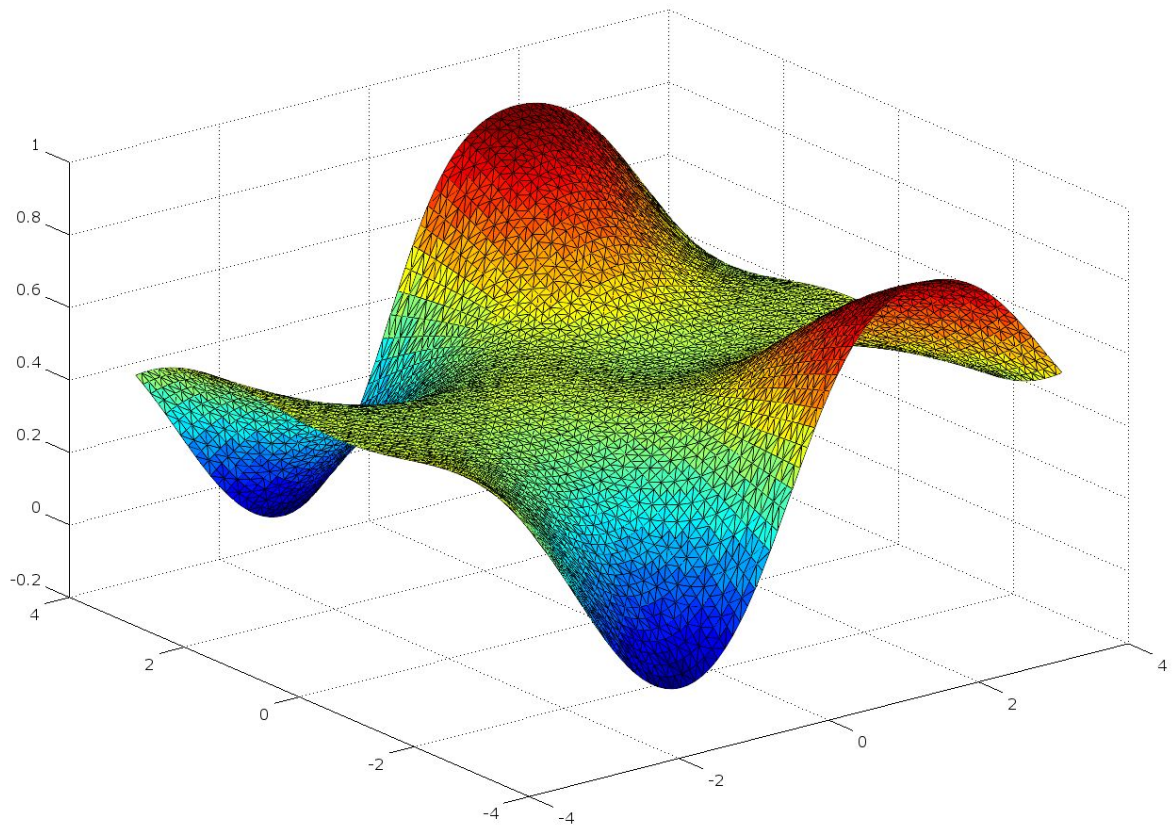


Figura 5: Terreno generado por la red neuronal.

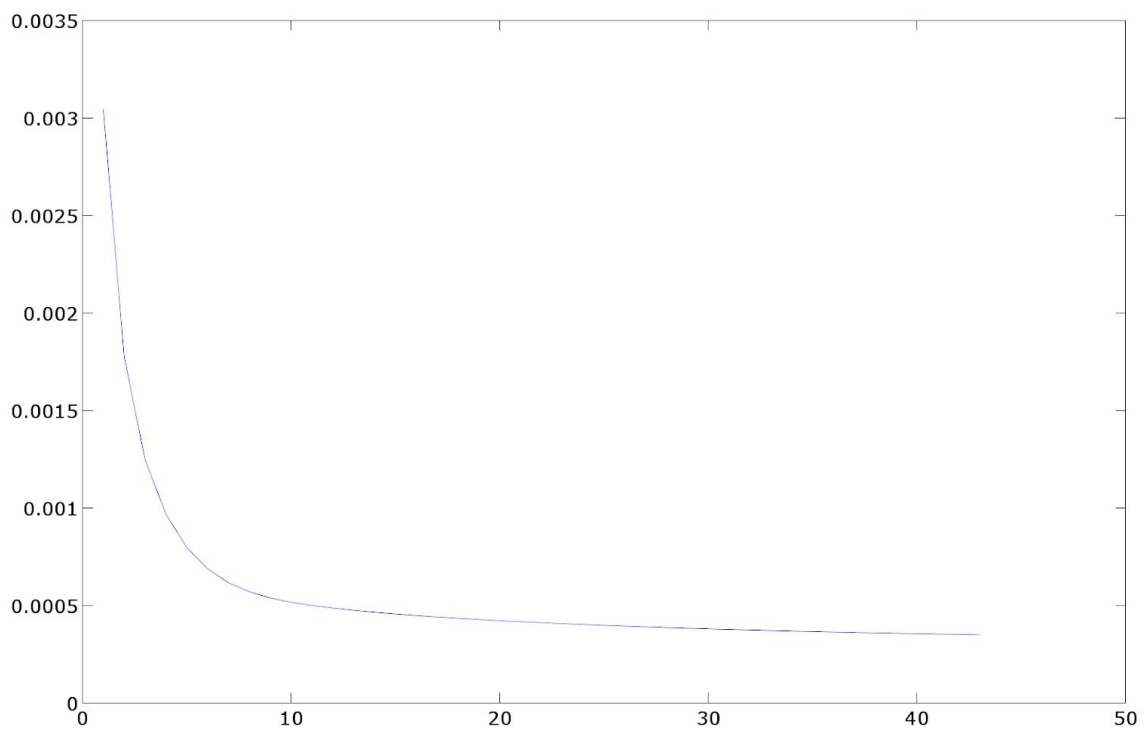


Figura 6: Error en función de las épocas en la arquitectura óptima.