

Sistemas de Inteligencia Artificial

Redes Neuronales

Trabajo Práctico Especial 2

Integrantes

52066 Soncini, Lucas
53214 Fraga, Matías
54449 Marzoratti, Luis
52051 De Lucca, Tomás

Profesores

Parpaglione, Maria Cristina
Pierri, Alan

1.Enunciado

Una empresa de videojuegos precisa un desarrollo que pueda simular en su plataforma terrenos de diferentes partes del mundo a partir de mediciones de altura, latitud y longitud. La red neuronal multicapa que se implemente deberá poder aproximar a la función de altura. Cada grupo debe construir los patrones de entrenamiento a partir del muestreo obtenido.

Para el desarrollo, deben considerarse los siguientes puntos:

- *Realizar un archivo de configuración, de forma tal que los aspectos relevantes de la red neuronal sean parametrizables.*
- *Utilizar como funciones de activación tanto la exponencial como la tangente hiperbólica y compararlas.*
- *Utilizar distintas arquitecturas de red y compararlas. Decir justificando en el informe con qué arquitectura aprendió mejor.*
- *Graficar la salida de la red neuronal de forma tal de analizar la función en el dominio que abarcan los puntos de entrada.*
- *Graficar el error (entrenamiento y testeo) de la red neuronal en su evolución.*
- *Obtener cualquier otra métrica que sea de utilidad para la defensa del funcionamiento de la red neuronal.*
- *Analizar la capacidad de generalización de la red.*
- *De las mejoras al algoritmo backpropagation implementar la mayor cantidad (por lo menos dos). Explicar cuáles fueron las elegidas. Comparar los resultados obtenidos con y sin estas mejoras.*
- *Considerar otras mejoras de optimización y precisión discutidas en clase.*

2.Introducción

Para la correcta resolución del problema propuesto se implementó una red neuronal multicapa, haciendo uso del algoritmo backpropagation visto en clase. Dentro de este marco se realizaron sucesivas pruebas posteriores a la implementación de dicho algoritmo

considerandos diversos parámetros a la hora de la construcción como se verá en apartados posteriores.

3. Toma de decisiones

Una de las decisiones importantes que se tomaron fue la de no dividir el conjunto de entradas en distintos conjuntos de testeo y entrenamiento, ya que se prefirió tomar la mayor cantidad de base de conocimientos disponible para aprender el terreno de la mejor manera.

Otra de las decisiones de implementación que se tomó fue la de no preocuparnos por la generalización, dicha decisión se encuentra fundada en la naturaleza de la consigna de igual manera que sucede en el caso anterior, el cual denota que la no importancia en la generalización viene fundada en hacer un entrenamiento de la mejor manera posible.

4. Mejoras

Para optimizar el proceso de *backpropagation*, ya que el mismo es computacionalmente intenso se evaluaron las siguientes mejoras:

ETA Adaptativo

La tasa de aprendizaje es la velocidad con la que se va aprendiendo. Al tener una tasa de aprendizaje alta (en el orden de 10^{-1}) el proceso de entrenamiento progresa con una mayor velocidad. Sin embargo, existe la posibilidad de que el algoritmo diverja. Si se opta por un *eta* bajo, se progresa lentamente hacia la solución pero este procedimiento puede llevar mucho tiempo.

Para mejorar esta situación se probó utilizar un *eta* adaptativo. Nuestra implementación de la optimización consiste analizar el cambio del error cuadrático medio a lo largo de 10 épocas y en función de esto modificar el valor de *eta* utilizado. Al ver que el error está aumentando en las últimas épocas (diverge) disminuimos el factor de aprendizaje un 10% y volvemos a la situación anterior. Por otro lado, cuando vemos que el error está disminuyendo en las últimas épocas, aumentamos el factor de aprendizaje un 10% para acelerar el proceso. De esta forma se puede aprovechar las mejores características de los distintos valores.

Momentum

Para las situaciones donde el gradiente descendente converge de una forma muy lenta o las oscilaciones son muy grandes, se puede introducir un término denominado

momentum. El objetivo de este término es homogeneizar las oscilaciones otorgando mayor importancia a la dirección de la pendiente.

En los casos probados en el presente trabajo se utilizó un $\alpha = 0.9$.

5.Arquitectura

Uno de los puntos más importantes a la hora de resolver el trabajo, fue el de determinar la configuración óptima que permitiese el mejor aprendizaje posible (siendo este, a nuestro criterio, el cual minimice el error y en segunda instancia tenga un tiempo de respuesta mínimo luego de haber sido entrenado).

En primer lugar quedó demostrado luego de las pertinentes pruebas que el aprendizaje era mejor (siendo esto, lo definido en el párrafo anterior) con una única capa oculta o con dos capas ocultas, arquitecturas que se verán en más profundidad en el próximo análisis. En sucesivas pruebas se obtuvieron errores similares utilizando dos y hasta tres capas y debido a que uno de los objetivos que buscamos es el de minimizar el tiempo de respuesta (no confundirlo con el tiempo que tarda en aprender la red), a errores similares, la implementación de arquitecturas de una capa sobresalió en los resultados, los cuales serán analizados en el próximo apartado.

Por otra parte, debido a que algunos puntos del conjunto se encontraban por fuera de la preimagen de la función, se decidió utilizar una transformación lineal en la salida y de esta manera poder contener a dichos puntos.

En concordancia con lo marcado anteriormente, la cantidad de neuronas que se encuentran en la capa oculta resultó ser elevado, en primer lugar porque un valor bajo hacía tender al error cuadrático medio a un valor alto, el cual como bien sabemos no podrá disminuir debido a que las neuronas se encontrarán saturadas, y en segundo lugar, tener varias capas más pequeñas en relación a una capa con una cantidad alta de neuronas, resultaba en un tiempo de respuesta considerablemente mayor, como se analizará en el apartado de resultados.

6.Resultados

Se probaron las siguientes arquitecturas de red:

- 1 capa oculta con 50 neuronas

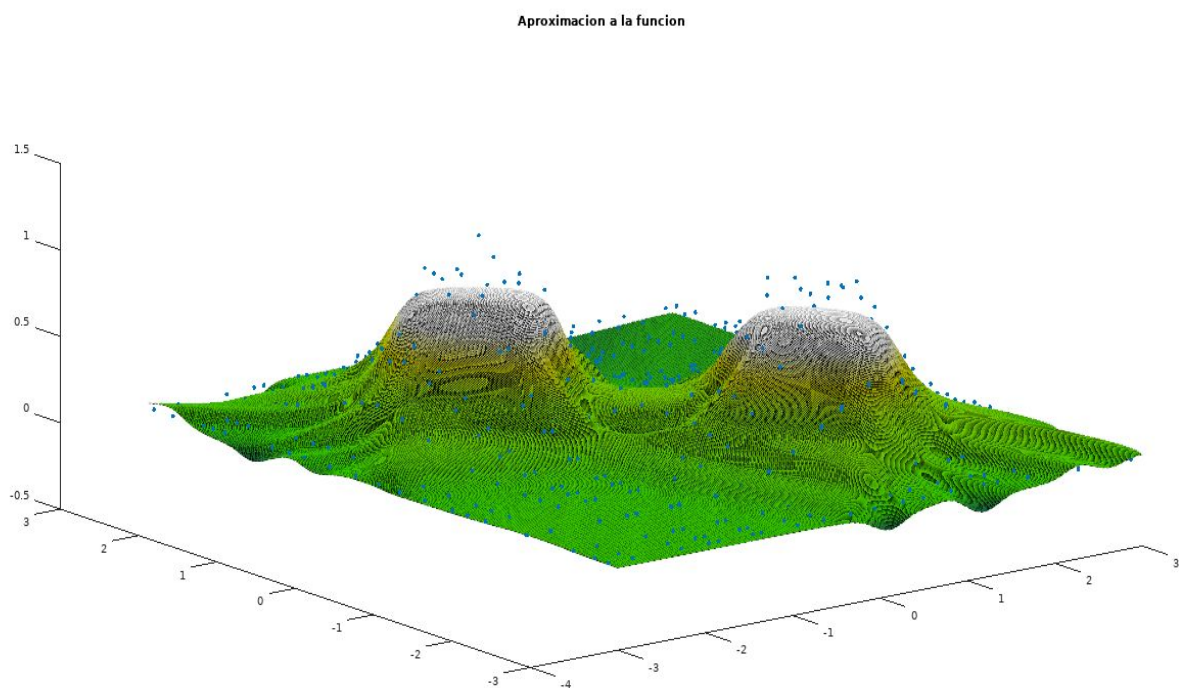
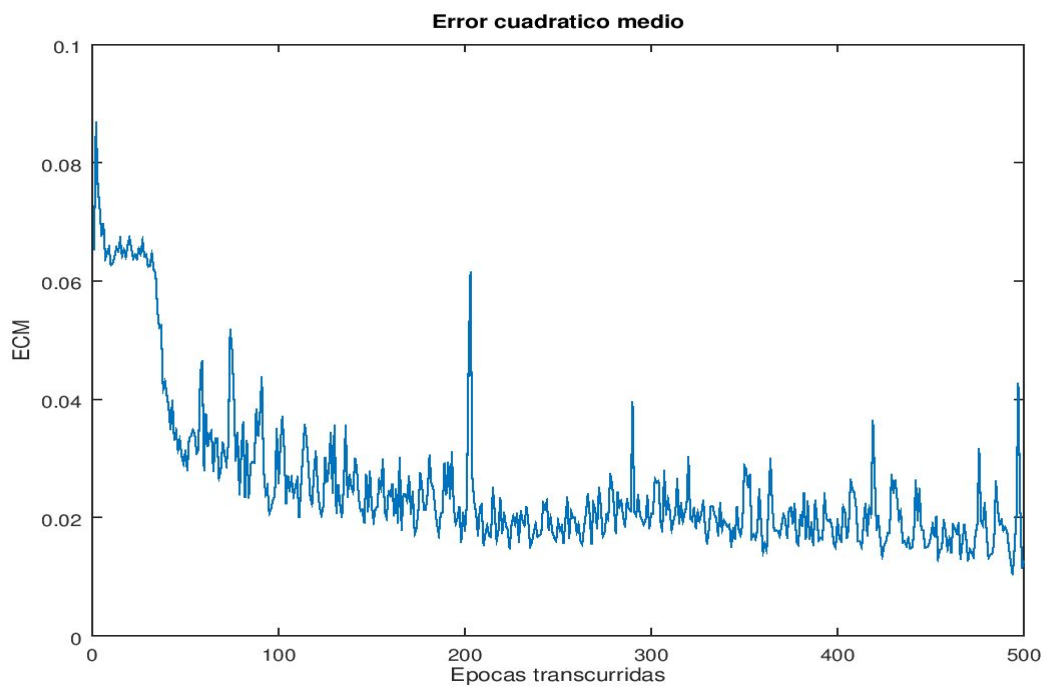
- 2 capas ocultas con 15 y 30 neuronas
- 2 capas ocultas con 45 y 50 neuronas

Épocas	Arquitectura	ECM	FA	B
200	2 15 30 1	$9,04 \cdot 10^{-4}$	Exponencial	15
200	2 50 1	$1,44 \cdot 10^{-3}$	Exponencial	15
200	2 45 50 1	$1,37 \cdot 10^{-4}$	Exponencial	15
200	2 15 30 1	$4,54 \cdot 10^{-4}$	Tanh	5
200	2 50 1	$1,77 \cdot 10^{-3}$	Tanh	5
200	2 45 50 1	$2,18 \cdot 10^{-4}$	Tanh	5

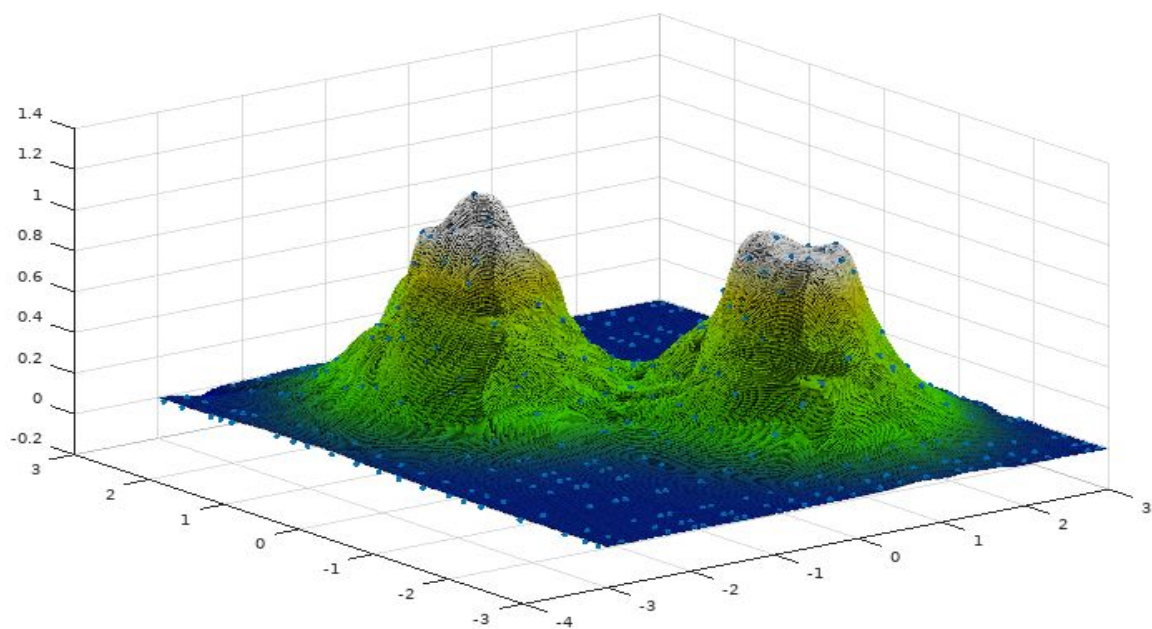
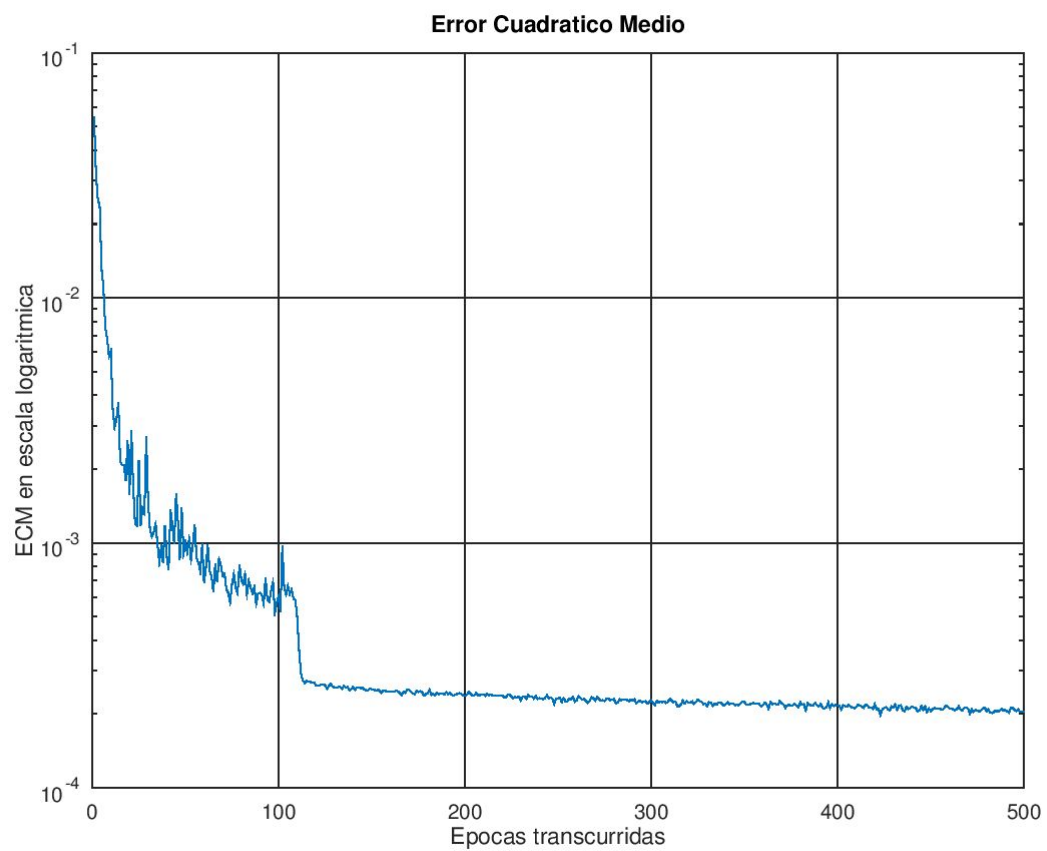
ECM	Arquitectura	Épocas	FA	Optimizaciones
10^{-2}	2 45 50 1	363	Exponencial	No
10^{-2}	2 45 50 1	83	Tanh	No
10^{-2}	2 45 50 1	146	Exponencial	Momentum
10^{-2}	2 45 50 1	71	Tanh	Momentum
10^{-2}	2 45 50 1	150	Exponencial	Eta adaptativo
10^{-2}	2 45 50 1	78	Tanh	Eta adaptativo

Anexo

Error cuadrático medio a lo largo de las primeras 500 épocas y aproximación de la función con $\eta = 0.01$, topología [2 45 50 1], con ambas optimizaciones, transcurridas 1000 épocas y con función de activación exponencial con $B = 1$.



Error cuadrático medio a lo largo de las primeras 500 épocas y aproximación de la función con $\eta = 0.05$, topología [2 45 50 1], sin optimizaciones, llegando a un error de 1.06×10^{-4} y con función de activación exponencial con $B = 15$.



Error cuadrático medio a lo largo de las primeras 500 épocas y aproximación de la función con la topología [2 45 50 1], con ambas optimizaciones, transcurridas 1000 épocas y con función de activación tangencial con $B = 1$.

