

# SLD124 FUNCIONES DE TRANSFERENCIA EN EL PERCEPTRÓN MULTICAPA: EFECTO DE SU COMBINACIÓN EN ENTRENAMIENTO LOCAL Y DISTRIBUIDO

## SLD124 TRANSFER FUNCTIONS IN THE MULTILAYER PERCEPTRON: EFFECTS OF ITS COMBINATION ON LOCAL AND DISTRIBUTED TRAINING

Yuleidys Mejias Cesar<sup>1\*</sup>, Ramón Carrasco Velaz<sup>2</sup>, Isbel Ochoa Izquierdo<sup>3</sup>, Edel Moreno Lemus<sup>4</sup>

<sup>1, 2, 3</sup> Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, Km 2½. Boyeros, Ciudad de La Habana, Cuba

\*Autor para la correspondencia: [ymejias@uci.cu](mailto:ymejias@uci.cu)

**RESUMEN:** *El perceptrón multicapa (PMC), figura dentro de los tipos de redes neuronales artificiales (RNA) que mejores resultados ha aportado en los estudios de relación estructura-actividad. Se conoce que los volúmenes de datos previstos para procesar en proyectos de Bioinformática son eventualmente grandes, por lo que se propuso evaluar algoritmos para acortar el tiempo de entrenamiento de la red sin afectar su eficiencia. Se evaluaron diferentes herramientas que trabajan con las RNA de las cuales se seleccionó el Weka para extraer el algoritmo de la red y la Plataforma de Tareas Distribuidas Tarenal para distribuir el entrenamiento del perceptrón multicapa. Finalmente se desarrolló un algoritmo para el entrenamiento local y distribuido del PMC con la posibilidad de variar las funciones de transferencias. Se demostró que en dependencia de la muestra de entrenamiento, la variación de las funciones de transferencias pueden reportar resultados mucho más eficientes que los obtenidos con la clásica función Sigmoideal, con incremento de la g-media entre el 4,5 y el 17%. Por otra parte se encontró que en los entrenamientos distribuidos es posible alcanzar eventualmente mejores resultados que los logrados en ambiente local.*

**Palabras claves:** funciones de transferencia, perceptrón multicapa, redes neuronales.

**ABSTRACT:** *The multilayer perceptron (PMC), ranks among the types of artificial neural networks (ANN), which has provided better results in studies of structure-activity relationship. It is known that the data volumes expected to process in Bioinformatics' projects are eventually big, so it was proposed to evaluate algorithms to shorten the training time the network without affecting its efficiency. It was evaluated different tools that work with ANN which was selected Weka algorithm for extracting the network and the Platform for Distributed Task Tarenal to distribute the training of multilayer perceptron. Finally it was developed a training algorithm for local and distributed the MLP with the possibility of varying transfer functions. It was shown that depending on the training sample, the change of transfer functions can yield results much more efficient than those obtained with the classic sigmoid function with increased g-media between 4.5 and 17%. Moreover, it was found that distributed training can be achieved eventually, better results than those achieved in the local environment.*

**Keywords:** transfer functions, Multi-layer perceptron, neural networks.

## 1. INTRODUCCIÓN

El presente trabajo surge en el marco del proyecto CITMA de investigación-desarrollo que se ejecutaba en la Facultad 6 de la Universidad de las Ciencias Informáticas, (UCI) denominado "Plataforma Inteligente para la Predicción de Actividad Biológica de Compuestos Orgánicos". En el aspecto de la modelación de la actividad biológica, las investigaciones en el proyecto se desarrollaban para el establecimiento de diferentes modelos matemáticos que describieran la actividad biológica de compuestos orgánicos, utilizando diferentes técnicas de Inteligencia Artificial (IA). Para describir la estructura química se partía de dos enfoques diferentes: división de la molécula en fragmentos ponderados por el Índice del Estado Refractotopológico Total y descriptores moleculares de tipo topológico, topográfico e híbridos, [1]. Como técnicas de IA se empleaban Lógica Difusa y Máquinas de Soporte Vectorial. No obstante, atendiendo a la naturaleza del problema y las capacidades de las Redes Neuronales Artificiales (RNA) de extraer información no-lineal de un conjunto de datos, se decidió ampliar las investigaciones en este campo para incrementar las potencialidades de la Plataforma en la búsqueda de modelos eficientes de relación estructura-actividad biológica.

Al enfrentar inicialmente el estudio de las RNA, se identificaron dos interrogantes fundamentales. Una de ellas estaba relacionada con el tiempo que requiere el entrenamiento de las redes y la capacidad de cómputo que consumen, más cuando se cuenta con grandes volúmenes de información, como es el caso de proyectos de Bioinformática. La otra interrogante estaba asociada a las funciones de transferencia de las redes y cómo su combinación podría afectar la eficiencia de la respuesta durante el entrenamiento de la red. La función de transferencia, o de activación, como también se le conoce, es uno de los elementos principales del funcionamiento de las RNA. La función de activación de una neurona artificial simula la respuesta de una neurona biológica ante un cierto estímulo. Se conoce que el cerebro posee un comportamiento diferente ante cada estímulo, sin embargo, en la mayoría de los casos las RNA son entrenadas y utilizadas haciendo uso de una única función de transferencia.

Por todo ello se trazó como objetivo determinar el efecto de la variación de las funciones de transferencia en la calidad de la respuesta del entrenamiento de un PMC, en ambiente local y distribuido.

El éxito de esta propuesta permitirá agilizar la etapa de desarrollo de nuevos fármacos, ya que a partir de la descripción de la estructura de una molécula, la red neuronal entrenada podrá predecir, con un porcentaje de error especificado por el usuario, la actividad biológica asociada a una molécula nueva a partir de sus descriptores.

## 2. DESARROLLO

### 2.1. Herramientas utilizadas

A partir del estudio de diferentes bibliografías, software y teniendo en cuenta las características del trabajo a desarrollar, se escogió Weka (Waikato Environment for Knowledge Analysis) para la extracción del algoritmo de las RNA, ya que es un software de código abierto que posee una colección de algoritmos de aprendizaje automático para la extracción de datos, [2]. Se escogió la Plataforma de Tareas Distribuidas (Tarenal) para realizar la distribución del entrenamiento ya que ofrece una alternativa de cómputo que aglutina en un solo conglomerado un conjunto de estaciones de trabajo, [3]. Se seleccionó Java como lenguaje de programación para realizar los cambios necesarios a los algoritmos presentes en el Weka y en la plataforma Tarenal y así ajustarlos al problema presente en esta investigación, puesto que es un lenguaje potente, con grandes capacidades de interconexión TCP/IP, por lo cual se puede acceder a la información disponible en red con mucha facilidad y seguridad.

Por otra parte se escogió Eclipse como herramienta IDE puesto que es un entorno de desarrollo integrado distribuido con el código abierto y multiplataforma. Se seleccionó el SPSS (Statistical Package for the Social Sciences) ya que es un software estadístico informático muy usado por las funcionalidades que brinda; el módulo utilizado en la presente investigación fue el de Pruebas no Paramétricas, el cual permite realizar distintas pruebas estadísticas especializadas en distribuciones no normales, [4]. El SPSS fue empleado con el objetivo de comparar los resultados de los 2 algoritmos utilizados, o sea, el algoritmo para el entrenamiento local de un perceptrón multicapa y el algoritmo distribuido, y de esta forma conocer si son significativas las diferencias entre las respuestas dadas por dichos algoritmos. Para ello se realizaron pruebas no paramétricas utilizando la medida de evaluación (g-media) de los algoritmos para realizar las comparaciones.

Se realizaron diferentes estudios sobre el funcionamiento y rendimiento del perceptrón multicapa, tanto en ambientes locales como distribuidos. Todos las pruebas o experimentos que se llevaron a cabo fueron realizados en computadores personales con un procesador Pentium 4 a 1.86 GHz de frecuencia y con 1Gb de RAM.

## 2.2 Arquitectura del perceptrón multicapa

La arquitectura del perceptrón multicapa se compone de: número de neuronas en la capa de entrada, el cual depende del número de componentes del vector de entrada; cantidad de capas ocultas y número de neuronas de cada una de ellas; número de neuronas en la capa de la salida, el cual depende del número de componentes del vector de salida o patrón objetivo. Teniendo en cuenta que tanto el vector de entrada como el de salida están definidos por el problema a resolver, no se hace difícil la determinación de la cantidad de neuronas en esas capas. La cantidad de neuronas en la capa de entrada depende de la cantidad de descriptores de la muestra escogida. En todos los casos, la cantidad de neuronas en la capa de salida será 1, ya que la función objetivo tendrá como salida una única respuesta (componentes activos o inactivos para las 2 primeras muestras y componentes de clase 1 o clase 2 para la última muestra).

Respecto al criterio a tener en cuenta para la selección de las neuronas de las capas ocultas surge una interrogante, este número en general debe ser lo suficientemente grande como para que se forme una región compleja que pueda resolver el problema, sin embargo no debe ser muy grande pues la estimación de los pesos puede ser no confiable para el conjunto de los patrones de entrada disponibles.

La mayor eficiencia del perceptrón multicapa en cuanto a su arquitectura se logra principalmente con una mayor experiencia de su diseñador, aunque también existen varios criterios para realizar esta selección.

A partir de una serie de criterios consultados en la bibliografía, se decidió trabajar con una sola capa de neuronas ocultas, ya que de esta manera se pueden representar todas las funciones matemáticas conocidas hasta el momento. [5]

Para determinar la cantidad adecuada de neuronas en las capas ocultas existen varios criterios, de ellos se seleccionó el que plantea que la cantidad de neuronas ocultas debe ser las dos terceras partes de la cantidad de neuronas de entrada más la cantidad de neuronas de salida. [5]

## 2.3 Entrenamiento del perceptrón multicapa

Inicialmente se fijaron algunos valores necesarios en el entrenamiento de la red como es el caso del momentum, con 0.001 y la velocidad de aprendizaje, con 0.003.

Luego de seleccionados los valores con los que se van a entrenar, y de definir la arquitectura de la red, la velocidad de aprendizaje y el momentum, además de la cantidad de patrones de entrada con que se va a realizar el entrenamiento, la aplicación genera aleatoriamente los pesos sinápticos.

Cada patrón de entrada se hace pasar a través de la estructura activando cada neurona y generando salidas en estas, dichas salidas son multiplicadas por los pesos sinápticos y constituyen la entrada de las neuronas de la capa siguiente, así sucesivamente hasta llegar a la capa de salida donde el resultado final es comparado con el resultado esperado, generando un error el cual es propagado por toda la red (backpropagation) hasta llegar al origen, corrigiendo los valores sinápticos. Así sucede con cada patrón hasta que todos hayan pasado a través de la red, esto constituye una iteración o *epoch*.

El criterio para establecer la condición de parada para el entrenamiento estuvo dado por la cantidad de iteraciones o epochs, o sea, el investigador define la cantidad de iteraciones que considere necesaria para obtener una red neuronal lo suficientemente entrenada para procesar sus datos de manera satisfactoria.

## 2.4 Muestras utilizadas durante la investigación. Características fundamentales.

Para realizar los estudios necesarios se utilizaron 3 muestras, dos de ellas de naturaleza bioinformática y que se utilizaron fundamentalmente para analizar el efecto que podría ocasionar la variación de las funciones de transferencia en la eficiencia de la respuesta de la red. Para realizar el entrenamiento estas muestras fueron divididas en 2 partes, el 80% para realizar el entrenamiento del perceptrón y el 20% restante para realizar las pruebas de la red entrenada. Las muestras se relacionan a continuación:

### Muestra 1: Cefalosporinas

Las cefalosporinas son compuestos antibacteriales pertenecientes a la familia de los  $\beta$ -lactámicos que poseen una gran actividad antibacterial. Estos antibióticos pueden utilizarse para el tratamiento de niños con fallos renales o hepáticos. [6]

La muestra utilizada contiene 179 descriptores y 104 instancias (o patrones de entrenamiento). Se conoce que todos los compuestos de la muestra son activos, es por ello que la investigación va enfocada a determinar los compuestos más activos y los menos activos. Para ello se tuvo que adaptar la variable dependiente (actividad biológica) para que pudiera ser tratada como un problema de clasificación. El procedimiento seguido fue el de determinar el promedio de los valores de la actividad biológica (AB) de cada compuesto y considerar como muy activos aquellos cuya AB fuera mayor o igual que el promedio, y como poco activos los que no cumplieran con este criterio. De esta manera la muestra quedó dividida en 52 elementos muy activos y 52 poco activos.

### **Muestra 2: Inhibidores del Factor 1 del receptor esteroideogénico (SF-1)**

La segunda muestra corresponde a los Inhibidores del Factor 1 del receptor nuclear esteroideogénico (SF-1). Este es un ensayo de dosis-respuesta basado en células para la inhibición del receptor huérfano. La muestra utilizada contiene 311 descriptores, 315 instancias (o patrones de entrenamiento) y la actividad biológica a predecir va a estar dividida en 175 elementos activos y 140 no activos.

### **Muestra 3: Reconocimiento facial**

La tercera muestra utilizada es de reconocimiento facial. Está destinada a la identificación automática de una persona en una imagen digital, mediante la comparación de determinadas características faciales a partir de una imagen digital o un fotograma de un video. A pesar de no ser de la rama de bioinformática se decidió trabajar con esta muestra por la cantidad de patrones que contiene. La muestra contiene 70 mil patrones de entrenamientos con 376 descriptores, cuenta con 63175 en la clase 1 y 6825 en la clase 2.

## **2.5 Reducción de variables**

Para realizar la reducción de variables se emplearon como métodos de búsqueda los Algoritmos Genéticos, el Enfriamiento Simulado y un Algoritmo Híbrido (compuesto por el Greedy y Algoritmos Genéticos). También se tuvieron en cuenta las medidas de evaluación CFS y Consistencia. La combinación de los métodos de búsqueda con las medidas de evaluación permitió obtener varias muestras de entrenamiento, las cuales fueron utilizadas para seleccionar las de entrenamiento.

## **2.6 Resultados y discusiones**

A continuación se muestran los resultados experimentales obtenidos, tanto en las pruebas realizadas en una computadora personal como en la plataforma de tareas distribuidas Tarenal. Se realiza una comparación de los resultados obtenidos en la calidad del modelo medido por la g-media, y el tiempo de ejecución del entrenamiento. También se comparan los valores de la g-media al entrenar las redes con diferentes funciones de transferencia.

### **2.6.1 Resultados experimentales del entrenamiento del perceptrón para la muestra de cefalosporinas**

De los 104 patrones de entrenamiento se utilizaron 83 para realizar el entrenamiento y los 21 restantes para desarrollar la validación. Se mantuvieron constantes los siguientes parámetros:

Cantidad de neuronas en la capa de entrada: 47  
Cantidad de neuronas en la capa oculta: 32  
Cantidad de neuronas en la capa de salida: 1  
Velocidad de aprendizaje: 0.003  
Momentum: 0.001  
Algoritmo de Búsqueda: Algoritmo Genético con 0.6 de probabilidad de cruzamiento y CFS como Medida de Evaluación.<sup>1</sup>

Los parámetros que se variaron fueron: cantidad de iteraciones y las funciones de transferencia utilizadas en la capa oculta y en la capa de salida. En la tabla I se muestran los resultados obtenidos.

---

<sup>1</sup> Yaikiel Hernández Díaz, Comunicación personal. Resultados de la Tesis de Maestría en Bioinformática, UCI, 2009

**Tabla I: Resultados experimentales de la muestra de cefalosporinas**

No	Cant de Iterac.	Funciones de transferencia Sigmoidal – Sigmoidal (a)		Funciones de transferencia TangenteSigmoidal- Sigmoidal (b)		Funciones de transferencia Lineal – Sigmoidal (c)	
		G-Media	Tiempo entrenam.	G-Media	Tiempo entrenam.	G-Media	Tiempo entrenam.
1	1000	0.7637	(0:0:9)	0.8606	(0:0:11)	0.8204	(0:0:9)
2	1500	0.8090	(0:0:14)	0.9045	(0:0:16)	0.0	(0:0:14)
3	2000	0.8606	(0:0:20)	0.9534	(0:0:22)	0.0	(0:0:19)
4	3000	0.8606	(0:0:30)	0.9534	(0:0:34)	0.0	(0:0:29)

  

No	Cant de Iterac.	Funciones de transferencia Sigmoidal- TangenteSigmoidal (d)		Funciones de transferencia Sigmoidal –Lineal (e)		Funciones de transferencia TangenteSigmoidal- TangenteSigmoidal (f)	
		G-Media	Tiempo entrenam.	G-Media	Tiempo entrenam.	G-Media	Tiempo entrenam.
1	1000	0.8204	(0:0:10)	0.8204	(0:0:10)	0.8606	(0:0:11)
2	1500	0.8606	(0:0:15)	0.9045	(0:0:15)	0.8606	(0:0:16)
3	2000	0.9045	(0:0:20)	0.9045	(0:0:21)	0.9045	(0:0:22)
4	3000	0.9045	(0:0:30)	0.9045	(0:0:30)	0.9045	(0:0:34)

(tabla III) que a pesar de que se clasificaron bien 7 de los 10 elementos muy activos, no se clasificó co-

**Tabla II: Matriz de confusión del ensayo 3-b**

Clase actual	Hipótesis	
	Negativos	Positivos
Negativos	VN= 10	FP= 1
Positivos	FN= 0	VP=10

**Tabla III: Matriz de confusión del ensayo 2-c**

Clase actual	Hipótesis	
	Negativos	Positivos
Negativos	VN= 7	FP= 11
Positivos	FN= 3	VP=0

Como se puede apreciar en la tabla I, el aumento del número de las iteraciones es directamente proporcional al aumento del tiempo (lo cual es bastante predecible), no siendo así con el valor de la G-Media, ya que en algunos casos este valor puede aumentar (aumenta la eficiencia de la respuesta del algoritmo, la red aprende más), puede disminuir (la red no aprende) o simplemente mantenerse constante.

De estos resultados se puede inferir que los mejores resultados de la red para la muestra analizada no se obtienen precisamente con la función de transferencia *Sigmoidal* en todas las capas, sino con la combinación de funciones *Tangente-Sigmoidal* en la capa oculta y *Sigmoidal* en la capa de salida, para la que se obtuvo un valor de G-Media igual a 0.9534. Se puede apreciar en la matriz de confusión asociada al ensayo 3-b (tabla II) que se solo se clasificó incorrectamente un valor del conjunto de validación.

Otro aspecto a resaltar es que para la muestra en cuestión no es conveniente utilizar la combinación de funciones *Lineal* (en la capa oculta) y *Sigmoidal* (en la capa de salida), ya que la red "no aprende". El valor de la G-Media igual a cero indica que la red no clasificó correctamente los patrones muy activos (+1) o ninguno de los pocos activos (-1) presentes en la muestra de validación. Se puede apreciar en la matriz de confusión asociada a este ensayo

rrrectamente ningún valor de los elementos poco activos.

Es importante destacar también el resultado obtenido con la combinación de funciones *Sigmoidal* y *Lineal*, ya que con solo 1500 iteraciones se obtuvo el segundo mejor valor de G-Media de las pruebas realizadas, al igual que con la combinación de funciones (b) del experimento 2.

Otro aspecto a observar es que en muchos casos no es necesario realizar muchas iteraciones de la red, ya que el mejor valor de la G-Media se logra con pocas iteraciones, como es el caso del ensayo 1-c.

### 2.6.2 Resultados experimentales del entrenamiento del perceptrón para la muestra de SF-1

Inicialmente se hizo un agrupamiento (clusterización) de la muestra debido a la diversidad estructural que poseía. De los 4 clúster resultantes se seleccionaron aleatoriamente los clústeres 0 y 3 para realizar las pruebas. A estos dos clústers se les aplicaron los algoritmos de búsqueda y las medidas de evaluación explicadas anteriormente para realizar la reducción del espacio muestral. En este caso se seleccionó el clúster 0 para realizar las pruebas a la red. De los 56 patrones de entrenamiento resultantes de la selección de

variables, se utilizaron 45 para realizar el entrenamiento y el 20% restante se utilizó para desarrollar la validación.

Para la realización de los experimentos se mantuvieron constantes las siguientes variables:

Cantidad de neuronas en la capa de entrada: 20

Cantidad de neuronas en la capa oculta: 14

Cantidad de neuronas en la capa de salida: 1

Velocidad de aprendizaje: 0.03

Momentum: 0.05

Algoritmo de Búsqueda: Enfriamiento Simulado con un valor de alfa igual a 0.7 y Consistencia como Medida de Evaluación.

Los parámetros que se variaron fueron: cantidad de iteraciones y las funciones de transferencias utilizadas en la capa oculta y en la capa de salida.

En la tabla IV se muestran los resultados obtenidos.

**Tabla IV: Resultados experimentales de la muestra de SF-1**

		Funciones de transferencias Sigmoidal – Sigmoidal (a)		Funciones de transferencias TangenteSigmoidal- Sigmoidal (b)		Funciones de transferencias Lineal – Sigmoidal (c)	
No	Cant de Iterac.	G-Media	Tiempo entrenam.	G-Media	Tiempo entrenam.	G-Media	Tiempo entrenam.
1	1000	0.8164	(0:0:1)	0.7745	(0:0:1)	0.7745	(0:0:1)
2	1500	0.7745	(0:0:1)	0.7745	(0:0:2)	0.7745	(0:0:1)
3	2000	0.7745	(0:0:2)	0.7745	(0:0:3)	0.8164	(0:0:2)
4	3000	0.7745	(0:0:3)	0.8164	(0:0:4)	0.8164	(0:0:3)
		Funciones de transferencias Sigmoidal- TangenteSigmoidal (d)		Funciones de transferencias Sigmoidal –Lineal (e)		Funciones de transferencias TangenteSigmoidal - TangenteSigmoidal (f)	
No	Cant de Iterac.	G-Media	Tiempo entrenam.	G-Media	Tiempo entrenam.	G-Media	Tiempo entrenam.
1	1000	0.6454	(0:0:1)	0.8660	(0:0:1)	0.7745	(0:0:1)
2	1500	0.5270	(0:0:1)	0.7745	(0:0:1)	0.8164	(0:0:2)
3	2000	0.5270	(0:0:2)	0.7745	(0:0:2)	0.8164	(0:0:2)
4	3000	0.5270	(0:0:3)	0.7745	(0:0:3)	0.8164	(0:0:4)

**Tabla V: Matriz de confusión del ensayo 1-e**

Clase actual	Hipótesis	
	Negativos	Positivos
Negativos	VN= 3	FP= 0
Positivos	FN= 2	VP= 6

**Tabla VI: Matriz de confusión del ensayo 2-d**

Clase actual	Hipótesis	
	Negativos	Positivos
Negativos	VN=1	FP= 1
Positivos	FN= 4	VP= 5

Al analizar los resultados obtenidos con la muestra SF-1 se puede apreciar que no son tan buenos como los resultantes de la muestra anterior (cefalosporinas). Se aprecia nuevamente cómo los mejores resultados de la G-Media se obtienen con combinaciones de funciones de transferencia distintas, que en este caso, a diferencia de los resultados de la muestra de cefalosporinas, se logra colocando la función *Sigmoidal* en la capa oculta y *Lineal* la en la capa de salida. El mejor valor de la G-Media obtenido fue de 0.8660 en el ensayo 1-e, al clasificar el 100 % de los valores

activos y el 60 % de los valores inactivos (Ver tabla V). El peor resultado se le atribuye a la combinación de Sigmoidal – Tangente Sigmoidal para el cual se obtuvo un valor de G-Media de 0.5270; en este caso se clasifican correctamente casi todos los valores activos, no siendo así con los valores inactivos (Ver tabla VI). Esta diferencia se le atribuye al desbalance presente en la muestra (2 elementos activos por cada 1 inactivo).

Por otro lado se aprecia que con todas las combinaciones de funciones de transferencia utilizadas la red aprende, no hay ningún caso en que el valor de la G-Media resultante sea cero.

### 2.6.3 Resultados experimentales del entrenamiento del perceptrón para la muestra de reconocimiento facial

De los 70 000 patrones de entrenamiento se utilizaron 60 000 para realizar el entrenamiento y los 10 000 restantes para desarrollar la validación. Se mantuvieron constantes los siguientes parámetros:

Cantidad de neuronas en la capa de entrada: 20

Cantidad de neuronas en la capa oculta: 14

Cantidad de neuronas en la capa de salida: 1  
 Velocidad de aprendizaje: 0.003  
 Momentum: 0.001  
 Los parámetros que se variaron fueron: cantidad de iteraciones y las funciones de transferencia

utilizadas en la capa oculta y en la capa de salida. En la tabla VII se muestran los resultados obtenidos.

**Tabla VII: Resultado experimental de la muestra 3 para el cálculo local y distribuido empleando la combinación de funciones de transferencias Sigmoideal- TangenteSigmoideal**

Entrenamiento Local				Entrenamiento Distribuido				
No	Cant de Iterac.	G-Media	Tiempo entrenam.	No	Cant de Iterac.	Cant. de PC	G-Media	Tiempo entrenam.
1	1000	0.6965	(0:27:36)	5	4000	2	0.6407	(1:15:00)
2	2000	0.7151	(0:53:3)	6	4000	4	0.6725	(0:45:00)
3	4000	0.6775	(1:45:20)	7	5000	2	0.6776	(1:30:23)
4	5000	0.6685	(2:16:34)	8	5000	4	0.6431	(0:55:00)

Al analizar los resultados mostrados en la tabla VII se puede apreciar que el valor de la G-Media en un entorno local se mantiene relativamente estable, inicialmente aumenta su valor y luego disminuye en cada iteración realizada, lo que demuestra que en este caso no es conveniente aumentar el número de iteraciones, pues la red no aprende más.

Como se puede apreciar en la tabla VII, el mejor valor de la G-Media que se obtiene es de 0.7151 y se logra en un entorno local.

Para conocer la eficiencia de la distribución del algoritmo para entrenamiento local se hicieron pruebas a los ensayos 3, 5 y 6 y al 4, 7 y 8.

En las pruebas realizadas a los ensayos 3, 5 y 6, al analizar el resultado reportado por el SPSS cuando se compararon los valores de la G-Media del algoritmo local y el distribuido, se obtuvo un valor

de 0.180, el cual revela que no es significativa la diferencia entre las respuestas dadas por los algoritmos. No obstante, no se recomienda hacer la distribución en 2 PC, sino en 4, ya que como se aprecia en el ensayo 5 la calidad de la respuesta para 2 PC es peor que la obtenida por el algoritmo en un entorno local, no siendo así en el caso de la distribución en 4 PC.

Al realizar las pruebas a los ensayos 4, 7 y 8 el valor obtenido con el test estadístico fue 0.655, el cual revela que tampoco hay diferencias significativas entre los resultados obtenidos. A pesar de ello no se recomienda hacer la distribución en más de 2 computadoras, ya que como se puede apreciar en el ensayo 8, el valor de la G-Media disminuyó al aumentar el número de iteraciones.

**Tabla VIII: Resultado experimental de la muestra 3 para el cálculo local y distribuido empleando la combinación de funciones de transferencias Sigmoideal-Sigmoideal.**

Entrenamiento Local				Entrenamiento Distribuido				
No	Cant de Iterac.	G-Media	Tiempo entrenam.	No	Cant de Iterac.	Cant. de pc	G-Media	Tiempo entrenam.
1	1000	0.6946	(0:25:46)	5	4000	2	0.6423	(1:10:00)
2	2000	0.6998	(1:9:42)	6	4000	4	0.5895	(0:44:21)
3	4000	0.6969	(1:51:34)	7	5000	2	0.6734	(1:31:00)
4	5000	0.6950	(2:20:27)	8	5000	4	0.5923	(0:57:35)

Los resultados de la tabla VIII muestran que a medida que aumenta el número de iteraciones en los entrenamientos realizados en un entorno local, no se reportan mejorías en el valor de la G-Media. El mejor resultado obtenido se logra con 2000 iteraciones, con un valor de G-Media de 0.6998.

Con la aplicación del test de Wilcoxon a los resultados de los ensayos 3, 5 y 6, se obtuvo un valor de 0.180, el cual revela que no es significativa

la diferencia entre los valores obtenidos en un ambiente local y el distribuido. Los resultados alcanzados sugieren que, para esta combinación de funciones, no es aconsejable hacer la distribución en más de 2 computadoras, ya que como se puede apreciar en la tabla 15, el valor de la G-Media disminuyó (ensayo 6). Además, debe valorarse cuidadosamente la distribución en 2 PC, ya que aunque el valor de la G-Media obtenido no

difiere considerablemente con el obtenido en un entorno local, la pérdida de dos unidades de precisión, puede resultar relevante para el resultado que se desea obtener.

Un resultado similar se obtiene al realizar las pruebas a los ensayos 4, 7 y 8 en que el valor obtenido con el test de Wilcoxon fue también de 0.180. A pesar de ello, tampoco se recomienda hacer la distribución en más de 2 computadoras, ya que como se puede apreciar, el valor de la G-Media disminuyó (ensayo 8).

Resulta interesante la diferencia entre los resultados obtenidos al aplicar diferentes combinaciones de funciones de transferencia a esta muestra. Solo con variar las combinaciones durante el entrenamiento de la red, se mejoraron los resultados en muchos casos, con lo que se demostró nuevamente el peso fundamental que tienen las funciones y sus combinaciones en la eficiencia de la respuesta de la red. En la actualidad no se cuenta con una respuesta definitiva de la razón por la cual algunas combinaciones de funciones reportan mejores resultados que otras, lo que será objeto de estudio en futuras investigaciones.

### 3. CONCLUSIONES

Con la realización del presente estudio:

- Se demostró que se reportan resultados diferentes con la variación de funciones de transferencia en el perceptrón multicapa, medibles por el valor de la G-Media. La sustitución de la combinación clásica Sigmoidea-Sigmoidea por la combinación de funciones Sigmoidea-Lineal o Lineal - Sigmoidea mejora los resultados en el orden del 4,5 al 17%.
- Se desarrollaron dos procedimientos que permiten realizar el entrenamiento del perceptrón multicapa en ambiente local y distribuido respectivamente, variando las funciones de transferencia para realizar el entrenamiento.
- Se disminuyó el tiempo de entrenamiento aproximadamente entre un 30-60% al entrenar la red en un ambiente distribuido.

El test estadístico aplicado sugirió que la calidad de los resultados en ambiente distribuido no se afecta de manera apreciable en comparación con los resultados en un entorno local.

### 4. REFERENCIAS BIBLIOGRÁFICAS

1. Escalona J.C, Carrasco R, Padrón J. A. Introducción al diseño de Fármacos. 2005. [Disponible en: <http://www.fq.uh.cu/investig/lqct/imagenes2/disen.pdf>]
2. Dimov Rossen. Weka: Practical Machine Learning Tools and Techniques with Java Implementations. 2007. [Disponible en: [http://www.dfki.de/~kipp/seminar\\_ws0607/reports/RossenDimov.pdf](http://www.dfki.de/~kipp/seminar_ws0607/reports/RossenDimov.pdf) ]
3. Proyecto Grid, Colectivo de Autores. *Manual del desarrollador - Plataforma de Tareas Distribuidas*. La Habana : s.n., 2009.
4. Departamento de Sociología IV. Introducción al análisis de datos. 2009. [Disponible en: [http://www.ucm.es/info/socivmyt/paginas/D\\_departamento/materiales/analisis\\_datosyMultivariable/19nparam\\_SPSS.pdf](http://www.ucm.es/info/socivmyt/paginas/D_departamento/materiales/analisis_datosyMultivariable/19nparam_SPSS.pdf) ]
5. Heaton Jeff. *Introduction to Neural Networks for Java, 2nd Edition*. St. Louis, Missouri : s.n., 2005. ISBN: 1604390085. 440p.
6. Carrasco R. Nuevos descriptores atómicos y moleculares para estudios de estructura-actividad. Aplicaciones. La Habana : Centro de Química Farmacéutica, Ministerio de Salud Pública, 2003. 114p.

### 5. SÍNTESIS CURRICULAR DE LOS AUTORES

**Yuleidys Mejias Cesar**, nació el 21 de junio de 1984 en Mayarí, Holguín. Realizó sus estudios pre-universitarios en el IPVCE José Martí, de Holguín. Sus estudios universitarios los cursó en la Universidad de las Ciencias Informáticas, donde se graduó en el año 2007 con título de oro. Discutió su maestría en Bioinformática en el año 2010 y en el 2012 se categorizó como profesora Asistente. Sus publicaciones han estado asociadas a la arquitectura de software y a la rama de la Bioinformática, específicamente en el campo de las redes neuronales artificiales.