



GRADO EN MATEMÁTICA COMPUTACIONAL

ESTANCIA EN PRÁCTICAS Y PROYECTO FINAL DE GRADO

---

# Aproximación de funciones con redes neuronales

---

*Autor:*  
Miguel AMELA LLOBET

*Supervisor:*  
David SÁEZ BAIXAULI  
*Tutor académico:*  
Juan José FONT FERRANDIS

Fecha de lectura: 28 de Noviembre de 2016  
Curso académico 2015/2016



## Resumen

Este documento trata sobre la Estancia en Prácticas en el Departamento de Informática del Hospital General de Castellón de la Plana y el Trabajo Fin de Grado de la asignatura *MT1030-Prácticas Externas y Proyecto Fin de Grado* del Grado en Matemática Computacional de la Universidad Jaume I.

El documento se divide en dos partes. En la primera, se describe la migración de un sistema web en Joomla a Wordpress, llevado a cabo durante la Estancia en Prácticas. En la segunda parte, se describe la capacidad para aproximar funciones continuas que tienen ciertas redes neuronales artificiales, y se muestran algunos ejemplos de redes que pueden realizar esta aproximación.

## Palabras clave

Red neuronal artificial, perceptrón multicapa, aproximación de funciones continuas, Teorema de Kolmogorov y Teorema de Stone-Weierstrass.

## Keywords

Neural network, multilayer perceptron, approximation of continuous functions, Kolmogorov Theorem and Stone-Weierstrass Theorem.



# Índice general

<b>1. Introducción</b>	<b>7</b>
<b>2. Estancia en Prácticas</b>	<b>9</b>
2.1. Introducción . . . . .	9
2.2. Objetivos del proyecto formativo . . . . .	9
2.3. Explicación detallada del proyecto realizado en la empresa . . . . .	9
2.4. Planificación temporal de las tareas . . . . .	11
2.4.1. Primera Quincena . . . . .	11
2.4.2. Segunda Quincena . . . . .	11
2.4.3. Tercera Quincena . . . . .	11
2.4.4. Cuarta Quincena . . . . .	13
2.4.5. Quinta Quincena . . . . .	13
2.4.6. Sexta Quincena . . . . .	14
2.5. Estimación de recursos del proyecto . . . . .	15
2.6. Grado de consecución de los objetivos propuestos . . . . .	15

<b>3. Memoria TFG</b>	<b>17</b>
3.1. Introducción . . . . .	17
3.2. Redes neuronales artificiales . . . . .	18
3.2.1. El perceptrón multicapa . . . . .	20
3.3. Aproximación de funciones utilizando redes neuronales . . . . .	21
3.3.1. Teorema de Kolmogorov . . . . .	21
3.3.2. Teorema de Stone-Weierstrass . . . . .	25
3.3.2.1. Red exponencial decreciente . . . . .	30
3.3.2.2. Redes de Fourier . . . . .	31
3.3.2.3. Red Sigma-Pi modificada o de polinomios . . . . .	32
3.3.2.4. Red con funciones exponenciales . . . . .	33
3.3.2.5. Red logística modificada . . . . .	34
<b>4. Conclusiones</b>	<b>35</b>

# Capítulo 1

## Introducción

Este documento presenta las tareas llevadas a cabo durante la Estancia en Prácticas y describe el Trabajo Fin de Grado, que trata sobre la capacidad de aproximar funciones continuas que presentan algunas redes neuronales artificiales.

En el segundo capítulo se detalla la Estancia en Prácticas en el Departamento de Informática del Hospital General de Castellón. El objetivo principal de estas prácticas era el de llevar a cabo una migración de un sistema web desarrollado en Joomla a la plataforma web Wordpress. Este sistema albergaba las páginas web de algunos centros de salud de la provincia de Castellón y los departamentos en los que se divide el Hospital General. La funcionalidad del nuevo sistema se debía mantener para que los usuarios de éste no rechazaran el cambio de la nueva plataforma encargada de gestionar dicho sistema. En este capítulo también se describen las tareas llevadas a cabo para configurar el servidor encargado de almacenar todo este sistema web.

En el tercer capítulo se presenta el Trabajo de Fin de Grado. En primer lugar se introducen las redes neuronales artificiales, las cuales intentan imitar la estructura y el comportamiento del cerebro humano. A continuación, se describe, desde un punto de vista teórico, la capacidad aproximativa de funciones continuas que presentan ciertas redes neuronales. Para ello, se estudian dos resultados clásicos del Análisis Matemático: los teoremas de Kolmogorov (1957) y de Stone-Weierstrass (1937).

Primero se prueba, aprovechando el Teorema de Kolmogorov, la capacidad aproximativa universal de las redes neuronales, tipo perceptrón. En la segunda parte del capítulo, se demuestra el Teorema de Stone-Weierstrass, resultado clave en el contexto de la aproximación de funciones, y se analizan qué hipótesis del citado teorema son satisfechas automáticamente por ciertas redes neuronales. Para finalizar este capítulo, se muestran algunos ejemplos de redes que poseen esta capacidad aproximativa como consecuencia del Teorema de Stone-Weierstrass.

En el último capítulo se enumeran las conclusiones que se derivan de la Estancia en Prácticas y del Trabajo Fin de Grado.



## Capítulo 2

# Estancia en Prácticas

### 2.1. Introducción

En este capítulo se desarrolla el proyecto que he llevado a cabo durante mi Estancia en Prácticas. La empresa en la que he desarrollado el proyecto es el Departamento de Informática del Hospital General Universitario de Castellón. Durante dichas prácticas, el supervisor que me han asignado ha sido David Sáez Baixauli, director del Servicio de Informática del citado Hospital, junto con Sara Querol.

### 2.2. Objetivos del proyecto formativo

El objetivo del proyecto a desarrollar durante las Estancia en Prácticas era el de realizar una migración de un sistema web integrado en la plataforma Joomla 1.5 a la plataforma Wordpress 4.5.2.

Tanto Joomla, como Wordpress, son plataformas web que sirven para crear y administrar cualquier tipo de sitio web, aunque principalmente se utilizan para albergar blogs.

### 2.3. Explicación detallada del proyecto realizado en la empresa

El sistema web que se debía migrar era el siguiente conjunto de páginas web:

- Comisión de calidad asistencial
- Comisión de docencia
- Centro de salud de Almazora
- Centro de salud Pintor Sorolla
- Centro de salud Rafalafena
- PAC (Punto de Atención Ciudadana) Benlloch - Vall d'Alba
- Dermatología
- Digestivo
- Medicina Preventiva
- UCI
- UDCA (Unidad de Documentación Clínica y Admisión)
- Urgencias
- Soporte
- DSCS (Departamento de Salud de Castellón)

De este conjunto, no era necesario migrar las páginas de Soporte y de DSCS. La de Soporte se debía suprimir. Por contra, en lugar de la de DSCS, se crearía una página con enlaces a las demás secciones.

Cada una de las páginas web tenía su propia instalación de Joomla 1.5. Esto quiere decir que cada una de estas secciones tenía su propio contenido. Por lo que era necesario realizar la migración de cada una de estas secciones.

Además, todas las secciones debían de estar bajo una misma instalación de Wordpress. De este modo, habría uno o varios administradores para cada sitio, y un superadministrador que se encargaría de gestionar todos los sitios de una forma global. Esto se consigue con Wordpress Multisite. Para ello se debe configurar esta opción después de la instalación.

## 2.4. Planificación temporal de las tareas

### 2.4.1. Primera Quincena

La primera parte del proyecto que se había de llevar a cabo consistía en elegir qué plataforma iba a albergar el nuevo sistema web. Las dos alternativas que se planteaban eran la de Joomla 3.5 o la de Wordpress 4.5.2. Ambas plataformas funcionan para crear y administrar cualquier tipo de sitio web.

Después de una búsqueda sobre qué plataforma se adaptaba mejor a las necesidades del sistema web, decidimos elegir Wordpress 4.5.2 como la plataforma que albergaría dicho sistema. El principal motivo de esta decisión era la facilidad de manejo que tiene Wordpress para la edición y creación de lugares web, ya que los futuros usuarios del sistema iban a ser médicos/as y enfermeros/as, por lo que debía tener un manejo sencillo.

### 2.4.2. Segunda Quincena

En este período me informé sobre las posibles formas de realizar la migración. La mayoría de opciones eran la utilización de un “plugin” (un complemento de Wordpress que aumenta su funcionalidad) dentro de la instalación de Wordpress, para que cogiera el máximo contenido posible para realizar la migración. El inconveniente de este método era que los menús, los ficheros descargables y las categorías con tilde en su nombre no los migraba, por lo que era necesario realizar la migración de este contenido de forma manual.

Una vez asimilado el proceso a seguir, instalé Wordpress 4.5.2 en el servidor local WampServer. A continuación, instalé los “plugins” candidatos a realizar la migración (CMS2CMS, Joomla/Mambo To WordPress Migrator y FG Joomla to WordPress). Estos “plugins” eran gratuitos y migraban contenido similar. Pero para ver que “plugin” se adaptaba mejor a la migración a realizar, probé los tres sobre la sección de Urgencias para ver realmente que contenido se migraba. Después de esta prueba, observé que el “plugin” que más contenido migraba era Joomla/Mambo To WordPress Migrator. Por este motivo decidí utilizar ese “plugin”.

### 2.4.3. Tercera Quincena

Durante esta quincena, utilicé el “plugin” Joomla/Mambo To WordPress Migrator para realizar la migración de la sección de Urgencias. A continuación, revisé todo el contenido que faltaba por migrar o corregir.

Faltaba corregir los nombres de las categorías en las que se agrupaban las entradas, crear los menús, subir los ficheros y revisar los usuarios.

Para llevar a cabo estas modificaciones, tuve que ver si con los “plugins” que trae por defecto Wordpress se podía hacer, ya que lo que se pedía era que el funcionamiento que debía tener fuera lo más cercano al funcionamiento que nos daba Joomla para que a los usuarios de esta nueva plataforma no vieran el cambio brusco de la migración.

Después de analizar cómo veía el usuario los menús, cómo se descargaban los ficheros, y compararlos con lo que ofrecía Wordpress, decidí instalar dos nuevos “plugins”: Wordpress Download Manager y Custom Menu Wizard Widget. El primero era para llevar a cabo la gestión de los ficheros y el segundo para crear los menús de una forma más parecida a la de Joomla. Además, para el primero tuve que instalar algunos complementos para aumentar su funcionalidad.

Después de esto, subí todos los ficheros descargables (pdf, doc, jpg, ppt, ...) que había en la plataforma de Joomla con el “plugin” Download Manager que acababa de instalar. Un problema con el que me encontré es que tuve que cambiar el tamaño máximo de los ficheros que permitía subir Wordpress, ya que estaba limitado a 2Mb y lo aumenté a 32Mb para poder albergar todos los documentos.

Una vez hecho esto, pasé a revisar los usuarios. Había que eliminar usuarios porque la base de datos de Joomla había sido hackeada y habían añadido usuarios falsos. Por lo tanto, era necesario hallar estos usuarios y eliminarlos.

A continuación pasé a modificar los nombres de las categorías que no se habían migrado correctamente, y sí en alguna categoría faltaba alguna entrada, la añadía.

Acto seguido, creé los menús. En primer lugar, hice los menús simples con las herramientas básicas de Wordpress. Utilicé el “plugin” de Custom Menu Wizard Widget para crear menús dinámicos de tal forma que al pulsar sobre un objeto del menú, desplegara su submenú en caso de que tuviera.

Después de esto, lo que hice fue tratar de mostrar las páginas tal cual se veían en Joomla. Tuve que instalar otro “plugin” (Simple Links) para gestionar los enlaces, ya que en Joomla se utilizan mucho y en Wordpress no se conseguía una funcionalidad parecida. Con este “plugin” añadía enlaces y además me permitía categorizarlos para realizar un posterior listado de enlaces por categorías. El “plugin” de Download Manager me permitía también hacer una selección de documentos por categorías. En último lugar, se debía modificar el aspecto visual del tema y de los “plugins” para que fuera parecido y tuviera consistencia. Para ello, lo que tuve que hacer fue modificar el estilo de Wordpress. Es decir, editar los ficheros de estilo (.css) que utilizaran el tema elegido y los “plugins”.

#### 2.4.4. Cuarta Quincena

Lo primero que hice durante esta quincena fue mostrarle a Sara Querol (mi supervisora junto con David Sáez) los avances en la migración. Me dijo que quería que la instalación de Wordpress 4.5.2 fuera “multisite”. Esto es, que bajo una misma instalación de Wordpress, pudiéramos tener varias webs independientes que serían las secciones anteriores. Por lo tanto, bajo un sitio principal, se podrían gestionar todos los demás sitios. Por lo tanto, reinstalé Wordpress y lo configuraré para que fuera “multisite”.

En la nueva instalación puse la sección de Urgencias que ya había realizado con anterioridad, con todo el contenido ya migrado.

Después de esto, lo que hice fue continuar con la migración. El siguiente sitio a migrar fue el de UDCA. Este sitio apenas tenía contenido por lo que hubo que hacerlo desde cero. Esto fue sencillo porque al establecer previamente un tema por defecto para todos los sitios, cuando se creaba un nuevo sitio incorporaba este tema de forma automática. El único inconveniente fue el de crear unos pequeños menús para que en un futuro tuviera una pequeña estructura para poder continuar a partir de ella con la creación de nuevo contenido web.

Por último, durante esta quincena, lo que hice fue migrar la sección de UCI. Este sitio ya tenía mucho más contenido, por lo que la migración fue más extensa que en el anterior caso. En primer lugar, utilicé el “plugin” de migración para migrar el contenido. En segundo lugar, modifiqué todas las categorías que no se habían migrado correctamente. A continuación, subí todos los documentos descargables que había en la antigua página de Joomla. Después de esto, creé todos los menús para dar acceso a todo el contenido del sitio, además de darle los mismos permisos de acceso y de qué contenido se podía descargar para la web.

En último lugar, había algunos usuarios que no se migraban. Después de investigar por qué no se llevaba a cabo la migración de algunos usuarios, llegué a la conclusión de que al poner todos los usuarios de Joomla en una misma base de datos (recordemos que en Joomla estaban en instalaciones diferentes, por lo tanto en bases de datos distintas) daba la casualidad de que algunos usuarios tenían el mismo nombre, por lo que un nuevo usuario con ese mismo nombre no se podía insertar en la base de datos.

#### 2.4.5. Quinta Quincena

En este periodo, la planificación fue la de configurar el nuevo servidor que albergaría la nueva red de sitios en la plataforma de Wordpress Multisite. Este servidor era un Windows Server 2008 R2.

En primer lugar, accedí de forma remota al servidor e instalé el entorno WampServer. A continuación lo configuré para que albergara, de una forma más óptima y con las funcionalidades deseadas, la plataforma web. La configuración realizada fue la siguiente:

- Habilitar los módulos de Apache *rewrite\_module*, *proxy\_module*, *proxy\_connect\_module*, *proxy\_http\_module*.
- Habilitar las extensiones de php *php\_openssl*, *php\_curl*, *php\_sockets*, *php\_openssl*.
- Habilitar en “settings” de php *file\_uploads*, *php\_sockets*, *php\_openssl*.
- Ampliar el tamaño de memoria de phpMyAdmin.
- Habilitar el inicio automático de WampServer al iniciar el servidor.
- Añadir una nueva norma en el “firewall” para el protocolo TCP por el puerto 80 para permitir conexiones.

En último lugar, creé la base de datos que albergaría todo el sistema web a través de phpMyAdmin. A continuación, instalé Wordpress y lo configuré para que fuera “multisite”. Después de esto importé la antigua base de datos (la que contenía los datos de la sección de UCI, UDCA y Urgencias) y los ficheros de cada sitio.

Una vez hecho todo esto, ya tenía en el servidor definitivo la migración de las tres secciones migradas hasta la fecha.

#### **2.4.6. Sexta Quincena**

Durante este periodo, escribí un manual de uso para el superadministrador de toda la red. Éste, además de servir como guía para la gestión de la propia plataforma de Wordpress Multisite, también incluye los pasos de la instalación realizada sobre el servidor, así como la configuración de dicho servidor. El motivo es que en un futuro se sepa todo lo que se hizo sobre ese servidor.

Durante la realización de este manual, continué con la migración de los demás sitios. Algunos de estos estaban vacíos por lo que solo había que crearlos.

El último día de esta quincena, realicé una presentación en la que mostraba la nueva plataforma web al equipo de Informática del Hospital General. En esta presentación también les enseñé a utilizar Wordpress Multisite para crear todo el contenido que en un futuro fuera necesario y el que iban a utilizar los médicos administradores de cada sitio.

Después de esta charla, Sara Querol y David Sáez me comentaron algunos aspectos a mejorar sobre la nueva plataforma. Entre éstos, había algunos relacionados con el aspecto que presentaba la página web y otros con la validación de usuarios por LDAP (Lightweight Directory Access Protocol); esto es, que para acceder a la web, valiera el mismo usuario que tiene el personal de cada sección para acceder a los ordenadores de los centros. Corregí lo del estilo de la web.

## **2.5. Estimación de recursos del proyecto**

Para el desarrollo del proyecto, tuve a mi disposición un ordenador de sobremesa con dos pantallas y con acceso a Internet. A través de este equipo realizaba una conexión remota con el servidor Windows Server 2003 que albergaba la plataforma web de Joomla y al servidor Windows Server 2008 R2 que albergaría la migración a Wordpress.

## **2.6. Grado de consecución de los objetivos propuestos**

El objetivo principal de la Estancia en Prácticas era el de realizar una migración de Joomla a Wordpress. Lo conseguí migrando el sistema web en Joomla 1.5 a Wordpress 4.5.2. La instalación de Wordpress pasó a ser “multisite”, por lo que se podía gestionar todo el contenido bajo una única instalación como se requería. Este contenido tenía el mismo diseño en todas las secciones.

Además de esto, también logré depurar la base de datos de los usuarios para dejar sólo los usuarios válidos, y por último conseguí realizar una configuración que facilitaría la creación de nuevos sitios web, ya que venía el tema por defecto.

Por otra parte, por falta de tiempo, no pude llevar a cabo la validación por LDAP, ni tampoco una optimización de Wordpress para evitar que la modificación simultánea de las secciones realizadas por varios administradores colapsara el servidor y por lo tanto, fuera algo más lento Wordpress.





## Capítulo 3

# Memoria TFG

### 3.1. Introducción

Una red neuronal artificial es un modelo matemático formado por una red interconectada de unidades simples, denominadas neuronas. Estas neuronas reciben unos datos de entrada, los analizan, y deciden qué hacer con ellos en base a la experiencia adquirida. Esto es, pasarlos a otras neuronas a través de las conexiones o directamente como salida aplicando algún tipo de función.

Las redes neuronales presentan las siguientes propiedades:

- Tienen la capacidad de realizar un aprendizaje adaptativo. Es decir, pueden realizar tareas a partir de un entrenamiento.
- Son auto-organizativas. Esto quiere decir que la red se organiza y se modifica con el fin de llevar a cabo un objetivo concreto.
- Tienen tolerancia a fallos, tanto para información de entrada incompleta, con distorsión, como para un fallo en ciertos nodos de la red. Esto es posible ya que existe cierto grado de redundancia de la información almacenada de la red neuronal.
- Realizan las operaciones en tiempo real. Debido a su implementación en paralelo, es posible ejecutarlas en paralelo, así se ahorra tiempo de entrenamiento y ejecución.

Las redes neuronales artificiales surgen de la necesidad de imitar el comportamiento de la mente humana a la hora de aprender y recordar cosas. Las redes neuronales artificiales se asemejan al funcionamiento de las redes neuronales biológicas.

Una red neuronal biológica está formada por neuronas (ver figura 3.1). En cada neurona, las dendritas son las encargadas de recibir un impulso eléctrico. Estos impulsos se suman y se convierten en una señal que es transmitida al soma. Si el valor de la señal proveniente de las dendritas es mayor que el valor umbral del cuello del axón, la señal se transmite al axón (neurona excitada). En caso contrario, no se transmite esta señal (neurona no excitada). Esta señal puede ir a otra neurona o directamente al sistema nervioso para responder a esta señal.

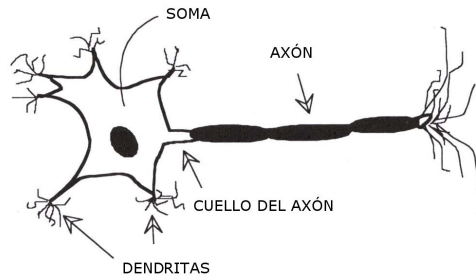


Figura 3.1: Neurona Biológica

### 3.2. Redes neuronales artificiales

En esta sección vamos a explicar el funcionamiento de una red neuronal artificial. Para ello vamos a explicar el funcionamiento de las neuronas que forman las redes neuronales (ver figura 3.2).

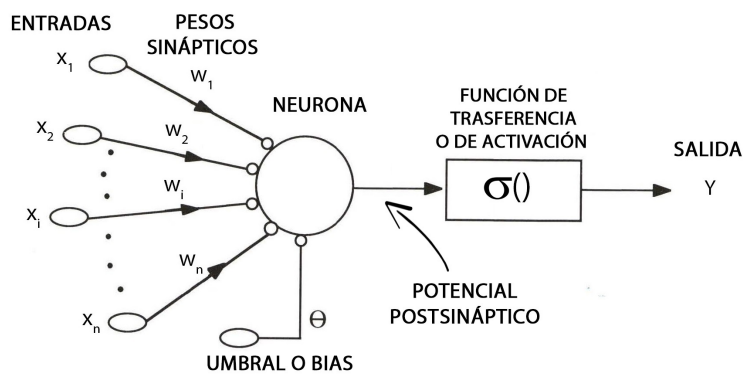


Figura 3.2: Neurona Artificial

Las entradas  $x_i$  son los datos que recibe la neurona para procesar. Cada entrada tiene un valor asociado  $w_i$ , que se conoce como peso sináptico. Este valor es un factor de importancia, que durante el entrenamiento de la red neuronal, varía para que tenga un mejor comportamiento. A continuación, podemos ver lo que es en sí la neurona o nodo. En ella calculamos un valor en función de las entradas y de los pesos, denominado potencial postsináptico.

Una operación muy utilizada para calcular el potencial postsináptico consiste en realizar el sumatorio de las entradas por los pesos

$$\sum_{i=1}^n x_i w_i$$

A continuación, un valor umbral  $\theta$  que indica a partir de qué valor del potencial postsináptico se activa la neurona o se produce una salida significativa. Por último, aparece la función de transferencia o de activación  $\sigma$  que se aplica a la diferencia entre el potencial postsináptico y el umbral, y devuelve el valor de salida de la neurona. Es decir, este valor es

$$\sigma \left( \sum_{i=1}^n x_i w_i - \theta \right)$$

Las funciones de activación más usadas son las sigmoideas:

**Definición 1** *Una función sigmoidea es una aplicación*

$$\sigma : \mathbb{R} \longrightarrow I(I = [0, 1])$$

tal que  $\lim_{t \rightarrow +\infty} \sigma(t) = 1$  y  $\lim_{t \rightarrow -\infty} \sigma(t) = 0$

Las funciones de activación sigmoideas más destacadas son las siguientes:

La función de Heaviside o escalón: 
$$\sigma(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$

La función sigmoide o logística: 
$$\sigma(x) = \frac{1}{1+e^{-x}}$$

La función lineal a trozos o rampa unidad: 
$$\sigma(x) = \begin{cases} 1 & \text{si } x \geq 1 \\ x & \text{si } 0 \leq x < 1 \\ 0 & \text{si } x < 0 \end{cases}$$

### 3.2.1. El perceptrón multicapa

En este apartado vamos a explicar una red neuronal a partir del perceptrón multicapa. Esta es una de las redes neuronales más conocida y usada. En este tipo de red la información fluye únicamente desde la entrada hasta la salida. Esto se conoce con el nombre de una red hacia delante (feedforward).

La arquitectura de la red se divide en la entrada, las capas ocultas y la salida (ver figura 3.3). En primer lugar tenemos la entrada de la red que son los datos que recibe la red para procesar y se conectan a las primeras neuronas de la primera capa oculta. Las capas ocultas son el conjunto de neuronas que no están conectadas directamente con la entrada. Estas neuronas están organizadas por capas y cada neurona de una capa, normalmente conecta con todas las neuronas de la capa siguiente sin formar ningún tipo de bucle. Esto se conoce como conectividad total.

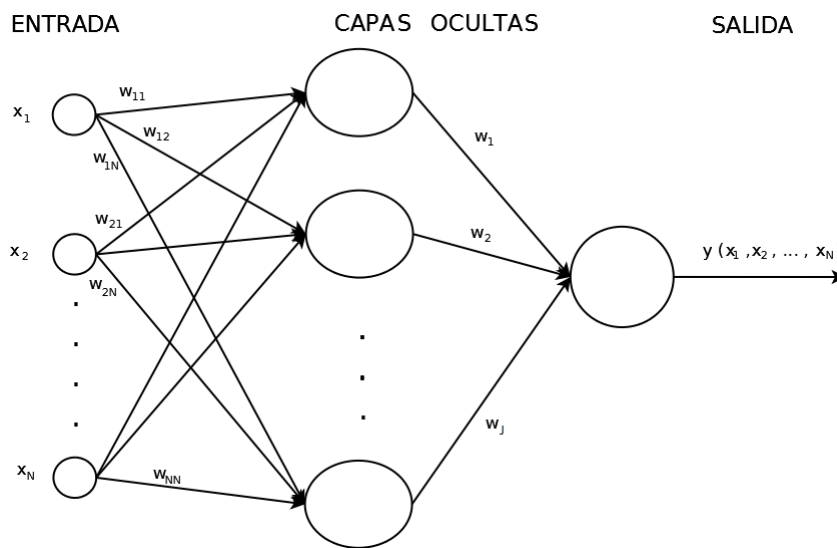


Figura 3.3: Perceptrón multicapa con una capa oculta y una unidad de salida. Es posible añadir más capas ocultas a la red y también más unidades de salida

El funcionamiento de cada neurona es el mismo que acabamos de ver en la figura 3.2. Por simplificar la representación en la figura 3.3, dibujamos la neurona juntamente con la función de transferencia. La función de transferencia que utiliza el perceptrón multicapa suele ser una función sigmoideal como las descritas anteriormente.

Por lo tanto la salida del perceptrón de la figura 3.3 queda de la siguiente forma:

$$y(x_1, x_2, \dots, x_N) = \sum_{j=1}^J w_j \sigma_j \left( \sum_{i=1}^N w_{ij} x_i + \theta_j \right) \quad (3.1)$$

siendo  $J$  el número de neuronas de la capa oculta,  $\sigma_j$  una función de transferencia sigmoideal,  $w_j$  el peso sináptico entre la neurona  $j$  y la neurona de la última capa y  $w_{ij}$  el peso sináptico entre la entrada  $i$  y la neurona  $j$  de la capa oculta.

### 3.3. Aproximación de funciones utilizando redes neuronales

En este apartado vamos a ver dos teoremas (Teorema de Kolmogorov y Teorema de Stone-Weierstrass) a partir de los cuales se obtienen las propiedades aproximativas de ciertas redes neuronales.

#### 3.3.1. Teorema de Kolmogorov

Vamos a empezar enunciando el Teorema de Kolmogorov, que demuestra que una función continua de  $n$  variables definida en el conjunto  $I = [0, 1]^n$  puede representarse como sumas y composiciones de un número finito de funciones de una sola variable.

Este teorema surge para ayudar en la solución del problema 13 de Hilbert, en el que se dice que las raíces de la ecuación

$$x^7 + ax^3 + bx^2 + cx + 1 = 0$$

como función de los coeficientes  $a$ ,  $b$  y  $c$  no son representables como sumas y composiciones de funciones ni siquiera de dos variables.

**Definición 2** Sea una función  $f : \mathbb{R} \rightarrow \mathbb{R}$ . Diremos que  $f$  es monótona creciente cuando, para todo  $x, y \in \mathbb{R}$  con  $x < y$  se cumple que  $f(x) \leq f(y)$ .

**Definición 3** Una función continua  $f : X \rightarrow Y$  definida entre los espacios topológicos  $X$  e  $Y$ , es universal si para cada función continua  $g : X \rightarrow Y$ , existe un punto  $x \in X$  tal que  $f(x) = g(x)$

**Teorema 1 (Kolmogorov 1957)** Para cada entero  $n \geq 2$  existen  $n \times (2n + 1)$  funciones monótonas crecientes  $\psi_{pq}$ , con  $p = 1, 2, \dots, n$  y  $q = 1, 2, \dots, 2n + 1$  con la siguiente propiedad:

Para cada función continua de valor real,  $f : I^n \rightarrow \mathbb{R}$ ; existen funciones continuas  $\phi_q$ ,  $q = 1, 2, \dots, 2n + 1$ , tales que

$$f(x_1, x_2, \dots, x_n) = \sum_{q=1}^{2n+1} \phi_q \left[ \sum_{p=1}^n \psi_{pq}(x_p) \right] \quad (3.2)$$

Las funciones  $\psi_{pq}$  son universales, y no dependen de  $f$  y las funciones  $\phi_q$  son continuas, de una sola variable y dependen de  $f$ .

Vamos a centrar nuestros resultados sobre perceptrones con una unidad de salida. Pasar a redes neuronales con varias unidades de salida es inmediato, ya que por la universalidad de las funciones  $\psi_{pq}$  del Teorema de Kolmogorov (Teorema 1), podemos conseguir una aplicación  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  a partir de  $m$  perceptrones multicapa con una sola unidad de salida, donde cada perceptrón tiene una función  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, m$ ; en este caso  $g(x) = (f_1(x), f_2(x), \dots, f_m(x))$ .

Vamos a analizar perceptrones multicapa y sus aplicaciones para la aproximación de funciones. Recordemos que este tipo de red neuronal calcula funciones de la forma

$$f(x) = \sum_{i=1}^k a_i \sigma(b_i \cdot x + c_i) \quad (3.3)$$

donde  $\sigma$  es una función de activación sigmoideal,  $a_i, c_i \in \mathbb{R}$ ,  $b_i, x \in \mathbb{R}^n$ ,  $k \in \mathbb{N}$ .

Veamos un teorema de aproximación derivado del Teorema de Kolmogorov (Teorema 1) y basado en las funciones anteriores (3.3), que muestra la capacidad aproximativa universal de ciertas redes neuronales:

**Definición 4** Diremos que una función  $f : \mathbb{R} \rightarrow \mathbb{R}$  es uniformemente continua cuando, para cada  $\epsilon > 0$ , podemos encontrar un  $\delta > 0$  tal que, si  $x, y \in \mathbb{R}$  con  $|x - y| < \delta$ , entonces  $|f(x) - f(y)| < \epsilon$

**Teorema 2 (Kůrková 1992)** Sean  $n \in \mathbb{N}$ ,  $n \geq 2$ ,  $\sigma : \mathbb{R} \rightarrow I$  una función sigmoideal continua,  $f \in C(I^n)$  y  $\epsilon$  un número real positivo. Entonces existen  $k \in \mathbb{N}$  y funciones  $\phi_i, \psi_{pq} \in S(\sigma)$  tales que para cada  $x = (x_1, x_2, \dots, x_n) \in I^n$  se cumple que

$$\left| f(x_1, x_2, \dots, x_n) - \sum_{i=1}^k \phi_i \left( \sum_{p=1}^n \psi_{pq}(x_p) \right) \right| < \epsilon$$

donde  $S(\sigma)$  es el conjunto de funciones del tipo (3.3)

*Demostración.* Por el Teorema de Kolmogorov (Teorema 1), sabemos que existen funciones continuas monótonas crecientes  $\psi_{pq}$  y funciones continuas  $\phi_q$  tales que

$$f(x_1, x_2, \dots, x_n) = \sum_{q=1}^{2n+1} \phi_q \left[ \sum_{p=1}^n \psi_{pq}(x_p) \right]$$

Sea  $[a, b] \subset \mathbb{R}$  tal que para  $p = 1, 2, \dots, n$ ,  $q = 1, 2, \dots, 2n + 1$  y  $\psi_{pq}(I) \subset [a, b]$ .

El conjunto de funciones  $f : [a, b] \rightarrow \mathbb{R}$ , con  $f(x) = \sum_{i=1}^k w_i \sigma(v_i x + u_i)$  donde  $\sigma : \mathbb{R} \rightarrow I$  es una función sigmoideal continua y  $w_i, v_i$  y  $u_i$  ( $i = 1, 2, \dots, k$ ) son números reales, es denso en  $C([a, b])$ .

Por ser denso, para cada  $q = 1, 2, \dots, 2n + 1$ , existe una función  $g_q \in S(\sigma)$  tal que

$$|g_q(x) - \phi_q(x)| < \frac{\epsilon}{(2n)(2n + 1)} \text{ para todo } x \in [a, b] \quad (3.4)$$

Debido a que las funciones  $g_q$  son uniformemente continuas, existe  $\delta > 0$  tal que

$$\text{si } |x - y| < \delta \text{ entonces } |g_q(x) - g_q(y)| < \frac{\epsilon}{(2n)(2n + 1)} \quad (3.5)$$

Además, para  $p = 1, 2, \dots, n$ , y  $q = 1, 2, \dots, 2n + 1$  existen  $h_{pq} \in S(\sigma)$  tales que

$$|h_{pq}(x) - \psi_{pq}(x)| < \delta \text{ para todo } x \in I \quad (3.6)$$

De (3.5) y (3.6) concluimos que

$$\left| \sum_{p=1}^n h_{pq}(x) - \sum_{p=1}^n \psi_{pq}(x) \right| < n\delta$$

y también

$$\left| \sum_{q=1}^{2n+1} g_q \left( \sum_{p=1}^n h_{pq}(x_p) \right) - \sum_{q=1}^{2n+1} g_q \left( \sum_{p=1}^n \psi_{pq}(x_p) \right) \right| < \frac{\epsilon}{2}$$

De (3.4) concluimos que

$$\left| \sum_{q=1}^{2n+1} \phi_q \left( \sum_{p=1}^n \psi_{pq}(x_p) \right) - \sum_{q=1}^{2n+1} g_q \left( \sum_{p=1}^n \psi_{pq}(x_p) \right) \right| < \frac{\epsilon}{2}$$

Por lo tanto,

$$\begin{aligned} & \left| f(x_1, x_2, \dots, x_n) - \sum_{q=1}^{2n+1} g_q \left( \sum_{p=1}^n h_{pq}(x_p) \right) \right| \\ &= \left| \sum_{q=1}^{2n+1} \phi_q \left( \sum_{p=1}^n \psi_{pq}(x_p) \right) - \sum_{q=1}^{2n+1} g_q \left( \sum_{p=1}^n h_{pq}(x_p) \right) \right| \\ &\leq \left| \sum_{q=1}^{2n+1} \phi_q \left( \sum_{p=1}^n \psi_{pq}(x_p) \right) - \sum_{q=1}^{2n+1} g_q \left( \sum_{p=1}^n \psi_{pq}(x_p) \right) \right| \end{aligned}$$



$$\begin{aligned}
& + \left| \sum_{q=1}^{2n+1} g_q \left( \sum_{p=1}^n \psi_{pq}(x_p) \right) - \sum_{q=1}^{2n+1} g_q \left( \sum_{p=1}^n h_{pq}(x_p) \right) \right| \\
& < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon \quad \blacksquare
\end{aligned}$$

### 3.3.2. Teorema de Stone-Weierstrass

Vamos a continuar con el que se considera el resultado más importante en la teoría de aproximación de funciones, el Teorema de Stone-Weierstrass. Este teorema es una generalización del clásico Teorema de Weierstrass (1885) sobre la densidad de los polinomios en el espacio de funciones reales continuas definidos sobre en un intervalo cerrado.

La generalización de Stone (1937) demuestra que las álgebras de funciones continuas son uniformemente densas en  $C(X)$  para cualquier  $X$  compacto, no necesariamente un intervalo cerrado. Recordemos que, si  $X$  es un espacio compacto de Hausdorff, denotamos por  $C(X)$  el conjunto de todas las funciones reales continuas en  $X$ .

Antes de empezar con el enunciado del teorema y su demostración, vamos a ver algunas definiciones, propiedades y proposiciones necesarias.

**Definición 5** Se dice que un espacio topológico  $(X, \mathcal{T})$  es de Hausdorff si dados dos puntos  $x$  e  $y \in X$  con  $x \neq y$  existen entornos disjuntos, es decir,  $\nu(x), \nu(y) \in \mathcal{T}$  tales que  $\nu(x) \cap \nu(y) = \emptyset$

**Definición 6** Se dice que un espacio topológico  $X$  es compacto si todo recubrimiento abierto de  $X$  admite un subrecubrimiento finito.

**Definición 7** Se dice que un espacio topológico  $X$  es normal si, y solo si, para cualquier par de cerrados disjuntos  $F$  y  $F'$ , existen dos entornos disjuntos,  $V$  y  $V'$ , tal que  $F \subset V$  y  $F' \subset V'$ .

**Definición 8** Un retículo es un conjunto  $L$  en el que cualquier conjunto finito tiene supremo e ínfimo y están en  $L$ . Si  $L$  es un retículo y  $x$  e  $y \in L$ , denotaremos  $x \vee y = \max[x, y]$ , y  $x \wedge y = \min[x, y]$ .

**Lema 1 (Urysohn 1924)** Si  $X$  es un espacio normal, entonces para cada par de subconjuntos cerrados y disjuntos  $A$  y  $B$  de  $X$  existe una función continua  $f : X \rightarrow [0, 1]$  tal que  $f(A) \subseteq \{0\}$  y  $f(B) \subseteq \{1\}$ .

## Lema 2

1. Sea  $X$  un espacio topológico normal. Por el Lema de Urysohn (Lema 1), existen suficientes funciones para separar puntos, esto es, que dados dos puntos distintos  $x$  e  $y \in X$ , podemos encontrar una función  $f$  en  $C(X)$  con  $f(x) \neq f(y)$ .
2. El espacio  $C(X)$  es un espacio vectorial porque cualquier función multiplicada por una constante es continua, y la suma de dos funciones continuas es continua.
3. El espacio  $C(X)$  es un espacio vectorial normado si definimos  $\|f\| = \sup|f(x)|$ , y un espacio métrico si tomamos  $\rho(f, g) = \|f - g\|$ .
4. El espacio  $C(X)$  tiene una estructura de anillo, ya que el producto  $f \cdot g$  de dos funciones  $f$  y  $g \in C(X)$ , está en  $C(X)$ .
5. Un subespacio vectorial  $A$  de  $C(X)$  es un álgebra si el producto de dos elementos de  $A$ , sigue estando en  $A$ .
6. El espacio  $C(X)$  tiene una estructura de retículo. Recordemos que un subespacio  $L$  de  $C(X)$  será un retículo si para cada par de funciones  $f$  y  $g \in L$  tenemos que  $(g \vee f)(x)$  y  $(g \wedge f)(x)$  están en  $L$  también.

**Proposición 1** Sea  $L$  un retículo de funciones continuas reales sobre un espacio  $X$  compacto, y supongamos que la función  $h(x) = \inf_{f \in L} f(x)$  es continua. Entonces, dado un  $\epsilon > 0$  podemos encontrar una función  $g \in L$  que cumpla que  $0 \leq g(x) - h(x) < \epsilon$  para todo  $x \in X$ .

*Demostración:* Para cada  $x \in X$  tenemos una función  $f_x \in L$  tal que  $f_x(x) < h(x) + \epsilon/3$ . Como  $f_x$  y  $h$  son funciones continuas, existe un abierto  $O_x$  que contiene a  $x$  tal que

$$|f_x(y) - f_x(x)| < \frac{\epsilon}{3} \quad \text{y} \quad |h(y) - h(x)| < \frac{\epsilon}{3} \quad \text{para todo } y \in O_x.$$

De las tres expresiones anteriores deducimos que  $f_x(y) - h(y) < \epsilon$  para todo  $y \in O_x$ . El conjunto de abiertos  $O_x$  cubre  $X$ . Entonces habrá un número finito de abiertos  $\{O_{x_1}, O_{x_2}, \dots, O_{x_n}\}$  que cubrirán  $X$  por ser  $X$  compacto. Sea  $g = f_{x_1} \wedge f_{x_2} \wedge \dots \wedge f_{x_n}$ . Por ser  $L$  un retículo,  $g \in L$ . Dado un  $y \in X$ , podemos elegir un  $i$  tal que  $y \in O_{x_i}$  y de esta forma tendremos que

$$g(y) - h(y) \leq f_{x_i}(y) - h(y) < \epsilon \quad \blacksquare$$

Ahora ya podemos enunciar el Teorema de Stone-Weierstrass para retículos.

**Teorema 3 (Stone-Weierstrass para retículos)** *Sea  $X$  un espacio compacto, y  $L$  un retículo de funciones continuas reales definidas sobre el compacto  $X$ , con las siguientes propiedades:*

- (1) *El retículo  $L$  separa puntos. Esto quiere decir que para cualesquiera  $x$  e  $y$  con  $x \neq y$ , existe una función  $f \in L$  que cumple que  $f(x) \neq f(y)$ .*
- (2) *Si tenemos una función  $f \in L$ , y una constante real  $c \in \mathbb{R}$ , entonces  $cf$  y  $c + f$  pertenecen a  $L$  también.*

*Entonces dada una función  $h \in C(X)$  y un  $\epsilon > 0$ , existe una función  $g \in L$  tal que para todo punto  $x \in X$  cumple que*

$$0 \leq g(x) - h(x) < \epsilon.$$

Antes de pasar a ver la demostración de este teorema, enunciaremos y probaremos dos lemas previos, que utilizaremos en la propia demostración del teorema.

**Lema 3** *Sea  $L$  un retículo de funciones reales sobre un espacio compacto  $X$  que satisface las propiedades (1) y (2) del Teorema 3. Entonces dados cualesquiera constantes reales  $a$  y  $b$  y cualesquiera par de puntos  $x$  e  $y$  distintos de  $X$ , existe una función  $f \in L$  tal que  $f(x) = a$  y  $f(y) = b$*

*Demostración:* Sea  $g$  una función en  $L$  con  $g(x) \neq g(y)$ . Sea

$$f(t) = \frac{(a-b)g(t)}{g(x)-g(y)} + \frac{bg(x)-ag(y)}{g(x)-g(y)}$$

Entonces  $f \in L$  por la propiedad (2), y sustituyendo en  $f$ , tenemos que  $f(x) = a$ ,  $f(y) = b$   $\blacksquare$

**Lema 4** Sea  $L$  un retículo de funciones reales sobre un espacio compacto  $X$  que cumple las propiedades (1) y (2) del Teorema 3. Sea  $a$  y  $b$  unas constantes reales con  $a \leq b$ . Sea  $F$  un conjunto cerrado de  $X$ , y  $p \notin F$ . Entonces existe una función  $f \in L$  con  $f(x) \geq a$ ,  $f(p) = a$  y  $f(x) > b$  para todo  $x \in F$ .

*Demostración:* Por el Lema 1 podemos elegir una función  $f_x$  con  $f_x(p) = a$  y  $f_x(x) = b + 1$ , para cada  $x \in F$ . Sea  $O_x = \{y : f_x(y) > b\}$ . El conjunto de  $\{O_x\}$  cubre  $F$ , y como  $F$  es compacto, podemos encontrar un número finito de abiertos  $\{O_{x_1}, O_{x_2}, \dots, O_{x_n}\}$  que cubren  $F$ . Sea  $f(x) = f_{x_1}(x) \vee f_{x_2}(x) \vee \dots \vee f_{x_n}(x)$ . Entonces  $f \in L$ ,  $f(p) = a$  y  $f(x) > b$  en  $F$ . Si ahora renombramos a  $f$  como  $f \vee a$  entonces nos aseguramos de que  $f \geq a$  en  $X$  ■

*Demostración del Teorema 3:* Como  $L$  es un conjunto no vacío, de la propiedad (2) sabemos que la función constante está en  $L$ . Por otro lado, definimos  $L' = \{f(x) : f(x) \in L \text{ y } f(x) \geq g(x)\}$  para  $x \in X$  y  $g \in C(X)$ . Si podemos demostrar que para cada  $p \in X$  tenemos que  $g(p) = \inf f(p)$  con  $f \in L'$ , esto es, la Proposición 1, entonces ya tendremos lo que buscamos.

Vamos a definir el conjunto  $F = \{x : g(x) \geq g(p) + \eta\}$  sienta  $\eta$  un número real positivo. Sabemos que  $F$  es cerrado, ya que  $g$  es continua. Además, como  $X$  es compacto,  $g$  está acotada en  $X$  por  $M$ , es decir  $g(x) \leq M$  para todo  $x \in X$ . Por el Lema 4, existe una función  $f \in L$  tal que  $f(x) \geq g(p) + \eta$ ,  $f(p) = g(p) + \eta$ , y  $f(x) > M$  en  $F$ . Como  $g(x) < g(p) + \eta$  en  $F$ , tenemos que  $f(x) > g(x)$  en  $X$ . Por lo tanto  $f \in L'$ , y  $f(p) \leq g(p) + \eta$ . Como este  $\eta$  habíamos dicho que es un número real positivo cualquiera, tenemos que  $g(p) = \inf f(p)$  con  $f(x) \in L'$  ■

Acabamos de ver los lemas, proposiciones y teoremas previos para ser capaces de demostrar el Teorema de Stone-Weierstrass, así que pasamos ahora a enunciarlo y posteriormente a demostrarlo.

A partir de la versión para retículos del Teorema de Stone-Weierstrass (Teorema 3) podemos obtener la versión más conocida de este Teorema; su versión para álgebras.

**Teorema 4 (Stone-Weierstrass 1937)** Sea  $X$  un espacio compacto y  $A$  un álgebra de funciones reales continuas definidas sobre  $X$ . Supongamos que  $A$  separa puntos en  $X$  y contiene las funciones constantes. Entonces dada cualquier función continua  $f \in C(X)$  y un  $\epsilon > 0$ , existe una función  $g \in A$  tal que para todo  $x \in X$  tenemos que  $|g(x) - f(x)| < \epsilon$ . Es decir,  $A$  es denso en  $C(X)$ .

*Demostración:* Si conseguimos probar que la clausura de  $A$  en  $C(X)$  es un retículo, por el Teorema 3, tendremos que ese  $A$  es denso en  $C(X)$ .

Sea  $\bar{A}$  la clausura de  $A \in C(X)$ . Entonces  $\bar{A}$  consiste en todas las funciones sobre  $X$  que tienen límite uniforme desde  $A$ . Sea  $f \in \bar{A}$  con  $\|f\| < 1$ . Entonces dado un  $\epsilon > 0$ , tenemos que

$\| |f(x)| - P(f(x)) \| < \epsilon$ , donde  $P(s)$  es un polinomio que vamos a describir.

Sea  $\sum_{n=0}^{\infty} c_n t^n$  la serie binomial para  $(1-t)^{1/2}$ . Esta serie converge uniformemente para un  $t \in [0, 1]$ ,

entonces podemos elegir un  $N$  que para todo  $t \in [0, 1]$  tengamos que  $\| (1-t)^{1/2} - Q_N(t) \| < \epsilon$

siendo  $Q_N(t) = \sum_{n=0}^{\infty} c_n t^n$ . Sustituyendo  $t$  por  $1 - s^2$  tenemos que  $\| |s| - Q_N(1 - s^2) \| < \epsilon$ . Si

definimos  $P(s) = Q_N(1 - s^2)$  tenemos que  $\| |s| - P(s) \| < \epsilon$ . Por lo tanto podemos deducir que  $P(f(x)) = Q_N(1 - f(x)^2)$ .

Como  $\bar{A}$  es un álgebra que contiene las funciones constantes,  $P(f(x)) \in \bar{A}$ . Además como  $\bar{A}$  es un conjunto cerrado de  $C(X)$ , tenemos que  $|f(x)| \in \bar{A}$ . Pero tenemos que

$$f \vee g = \frac{1}{2}(f + g) + \frac{1}{2}|f - g|$$

$$f \wedge g = \frac{1}{2}(f + g) - \frac{1}{2}|f - g|$$

Entonces  $\bar{A}$  es un retículo y, por el Teorema 3,  $\bar{A}$  es  $C(X)$  ■

El resto de esta sección está dedicada al estudio de algunas redes neuronales que satisfacen las hipótesis del Teorema de Stone-Weierstrass.

Estas redes tienen normalmente forma de árbol, y están estructuradas en varias capas, teniendo una única salida final (realizando  $N$  copias de la red, podríamos obtener  $N$  salidas). En las capas ocultas de la red, cada neurona tiene una única salida que va hacia otra neurona. Por último, las funciones de activación que utilizan son funciones sigmoideas.

Veamos cómo se comportan, en general, las redes neuronales en relación a las hipótesis del Teorema de Stone-Weierstrass.

La primera hipótesis requiere que la red neuronal sea capaz de albergar la función identidad  $f(x) = 1$ . Una forma de ver esto es poner todos los pesos sinápticos de la red neuronal a 0, excepto el peso de una entrada de la última neurona. De esta forma aseguramos que la salida de la red neuronal es 1.

La segunda hipótesis es la de separabilidad de puntos. La red neuronal ha de ser capaz de procesar funciones que separen puntos, es decir, que en cada par de puntos, la red tome valores distintos. Esta hipótesis se cumple si la red neuronal es capaz de procesar una función estrictamente monótona por cada variable de entrada. Cualquier red neuronal del tipo anteriormente citado tiene esta propiedad.

La tercera hipótesis del Teorema de Stone-Weierstrass requiere que la red neuronal sea capaz de aproximar sumas y productos por constantes de funciones. Si tenemos una red neuronal que puede aproximar las funciones  $f$  y  $g$ . Entonces, también podrá aproximar  $af + bg$ . Para ello construimos una red que en la penúltima capa tenga dos salidas, siendo éstas  $f$  y  $g$ . Entonces unimos estas salidas a una neurona, siendo los pesos de unión  $a$  y  $b$ , y la salida de la red será  $af + bg$ .

La cuarta hipótesis que debe satisfacer la red neuronal ha de ser la de procesar la multiplicación de funciones  $fg$ . Como asumimos que la salida de la red neuronal será lineal, entonces debemos expresar el producto  $fg$  como suma de funciones. La clave está en hallar funciones que transforman multiplicaciones en sumatorios. Esto se consigue tomando como funciones de transferencia funciones exponenciales, fracciones parciales o funciones escalón, por ejemplo.

A continuación vamos a ver algunas redes neuronales que cumplen las hipótesis del Teorema de Stone-Weierstrass.

### 3.3.2.1. Red exponencial decreciente

Este tipo de redes utilizan una función de activación exponencial, gracias a la cual, podemos transformar multiplicaciones en sumas.

**Corolario 1** Sea  $\mathfrak{S}$  el conjunto de funciones definidas en el compacto  $D = [0, 1]^N$  que pueden ser calculadas por una red neuronal exponencial decreciente, es decir:

$$\mathfrak{S} = \left\{ f(x_1, x_2, \dots, x_N) = \sum_{i=1}^I w_i \exp \left( - \sum_{n=1}^N w_{in} x_n \right) : w_i, w_{in} \in \mathbb{R} \right\}$$

Entonces  $\mathfrak{S}$  es denso en  $C(D)$

*Demostración:* Por lo comentado anteriormente, basta comprobar la cuarta hipótesis del Teorema de Stone-Weierstrass, es decir, vamos a ver que el producto de dos funciones está en  $\mathfrak{S}$ .

Sean  $f$  y  $g$  dos funciones de  $\mathfrak{S}$  de la forma:

$$f(x_1, x_2, \dots, x_N) = \sum_{i=1}^I w_i \exp \left( - \sum_{n=1}^N w_{in} x_n \right), \quad g(x_1, x_2, \dots, x_N) = \sum_{j=1}^J w_j \exp \left( - \sum_{n=1}^N w_{jn} x_n \right)$$

Entonces el producto de  $fg$  puede escribirse como un solo sumatorio:

$$fg = \sum_{k=1}^{IJ} w_k \exp\left(-\sum_{n=1}^N w_{kn}x_n\right), \text{ siendo } w_k = w_iw_j \text{ y } w_{kn} = w_{in} + w_{jn}.$$

Por lo tanto  $fg \in \mathfrak{S}$  y la red neuronal satisface las hipótesis del Teorema de Stone-Weierstrass. Entonces  $\mathfrak{S}$  es denso en  $C(D)$  ■

A continuación describimos varias redes neuronales que también satisfacen las hipótesis del Teorema de Stone-Weierstrass. La comprobación de las hipótesis es similar a la del caso anterior.

### 3.3.2.2. Redes de Fourier

Las redes de Fourier son las redes que implementan series de Fourier, las cuales satisfacen el Teorema de Stone-Weierstrass ya que por propiedades trigonométricas, podemos transformar multiplicaciones en sumas, como podemos ver en el siguiente ejemplo:

$$\cos(A + B) = \cos(A) \cos(B) - \sin(A) \sin(B)$$

La función de transferencia de la red de Fourier se obtiene haciendo algunas modificaciones a la función *Coseno* y obteniendo la función  $\psi$

$$\psi = \begin{cases} 0 & \text{si } x \leq -\frac{\pi}{2} \\ \frac{1 + \cos(\frac{3\pi}{2} + x)}{2} & \text{si } -\frac{\pi}{2} < x < \frac{\pi}{2} \\ 1 & \text{si } x \geq \frac{\pi}{2} \end{cases}$$

Entonces la red de Fourier se define de la siguiente manera:

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^I w_i \psi\left(\sum_{n=1}^N w_{in}x_n + \theta_i\right)$$

Otra red que se deriva de la red de Fourier es la red del *Cosig*. El nombre se debe a la función de transferencia, que surge de una modificación en la función de transferencia que utiliza la red de Fourier, pero acortando su dominio. Esta función queda de la siguiente manera:

$$\text{cosig}(x) = \begin{cases} 0 & \text{si } x \leq -\frac{1}{2} \\ \frac{1 + \cos(2\pi x)}{2} & \text{si } -\frac{1}{2} < x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$$

Entonces la red del *Cosig* tiene la siguiente forma.

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^I w_i \text{cosig} \left( \sum_{n=1}^N w_{in} x_n + \theta_i \right)$$

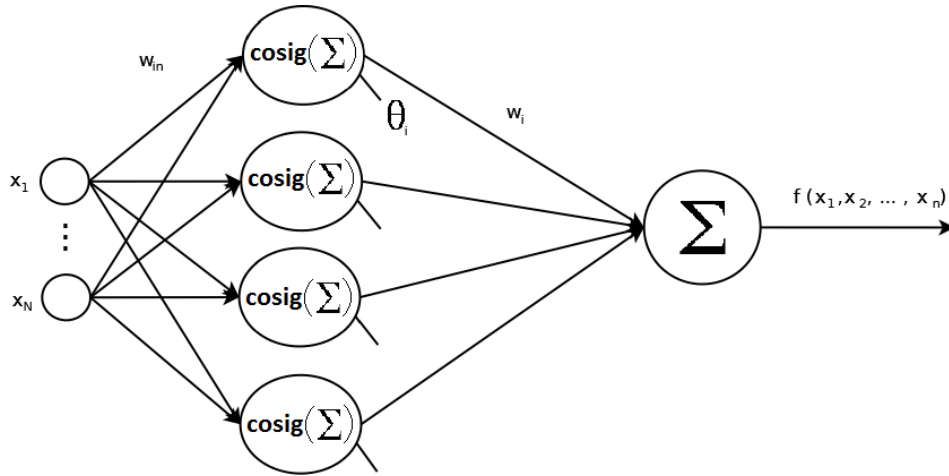


Figura 3.4: Red del *Cosig* con estructura en forma de árbol. En la primera capa aplicamos la función *Cosig* a la suma de las entradas por sus respectivos pesos, menos  $\theta$ . En la última neurona sumamos las entradas multiplicadas por sus respectivos pesos.

**Corolario 2** Sea  $\mathfrak{S}$  el conjunto de todas las funciones definidas en el compacto  $D = [0, 1]^N$  que pueden procesar una red de Fourier o una red del *Cosig*. Entonces  $\mathfrak{S}$  es denso en  $C(D)$ .

### 3.3.2.3. Red Sigma-Pi modificada o de polinomios

Este tipo de red se basa en funciones exponenciales también. Tienen la capacidad de aproximar funciones de la forma  $x_1^{q_1} x_2^{q_2} \cdots x_n^{q_n}$  siendo  $q_i \in \mathbb{R}$ . En la red que vamos a ver, los pesos sinápticos funcionan como exponente del polinomio. Además, si intercambiamos las funciones de las neuronas, tendríamos un resultado similar.



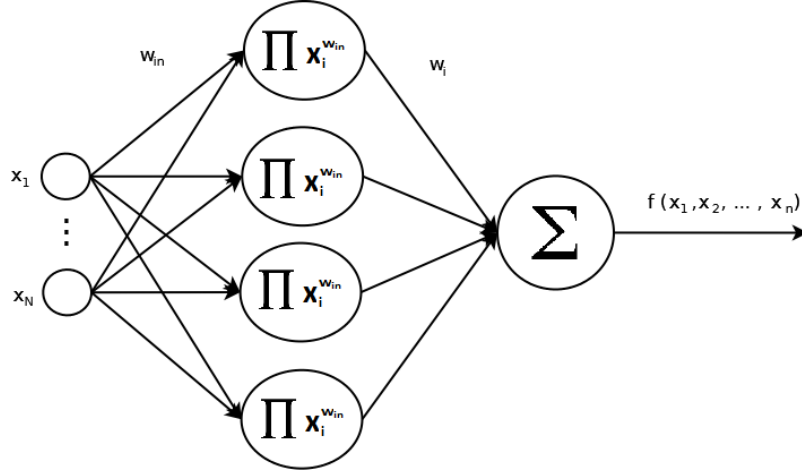


Figura 3.5: Red Sigma-Pi modificada con estructura en forma de árbol. En la primera capa se multiplican las entradas elevándolas a sus respectivos pesos, y en la neurona de la última capa, se suma cada entrada multiplicada por su respectivo peso.

**Corolario 3** Sea  $\mathfrak{S}$  el conjunto de todas las funciones definidas en el compacto  $D = [0, 1]^N$  que pueden procesar una red sigma-pi, es decir:

$$\mathfrak{S} = \left\{ f(x_1, x_2, \dots, x_n) = \sum_{i=1}^I w_i \prod_{n=1}^N x_n^{w_{in}} : w_i, w_{in} \in \mathbb{R} \right\}$$

Este conjunto es denso en  $C(D)$ .

### 3.3.2.4. Red con funciones exponenciales

Este tipo de red se basa en las funciones exponenciales también. Podemos obtener esta red añadiendo una función  $g(x)$  a las entradas de una red sigma-pi modificada. Esta función  $g(x)$  ha de ser invertible.

**Corolario 4** Sea  $\mathfrak{S}$  el conjunto de todas las funciones definidas en el compacto  $D = [0, 1]^N$  que pueden procesar una red con funciones exponenciales:

$$\mathfrak{S} = \left\{ f(x_1, x_2, \dots, x_n) = \sum_{i=1}^I w_i \prod_{n=1}^N g(x_n)^{w_{in}} : w_i, w_{in} \in \mathbb{R}, g \in C([0, 1]) \text{ invertible} \right\}$$

Este conjunto es denso en  $C(D)$ .

### 3.3.2.5. Red logística modificada

Este tipo de red deriva de las redes exponenciales. Utiliza como función de transferencia la función logística.

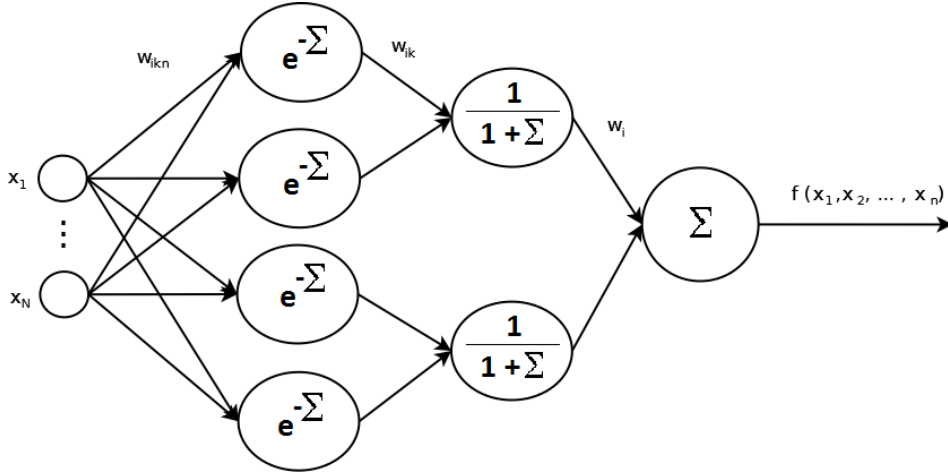


Figura 3.6: Red logística modificada con estructura en forma de árbol. En la primera capa elevamos negativamente el número  $e$  al sumatorio de las entradas multiplicadas a sus respectivos pesos. En la siguiente capa sumamos las entradas multiplicadas por sus respectivos pesos, le sumamos 1 y elevamos el resultado a  $-1$ . En la última capa realizamos el sumatorio de las entradas multiplicadas por sus respectivos pesos.

**Corolario 5** Sea  $\mathfrak{S}$  el conjunto de todas las funciones definidas en el compacto  $D = [0, 1]^N$  que pueden procesar una red logística modificada:

$$\mathfrak{S} = \left\{ f(x_1, x_2, \dots, x_n) = \sum_{i=1}^I w_i \left[ 1 + \sum_{k=1}^K \exp \left( - \sum_{n=1}^N w_{ikn} x_n - \theta_{ik} \right) \right]^{-1} : w_i, w_{ikn}, \theta_{ik} \in \mathbb{R} \right\}$$

Este conjunto es denso en  $C(D)$ .

## Capítulo 4

# Conclusiones

Durante la Estancia en Prácticas realicé una migración de un sistema web de una plataforma a otra de forma satisfactoria. Pude observar toda la problemática asociada a intentar que la funcionalidad de este sistema web fuera la misma independientemente de la plataforma web seleccionada. Aprendí a resolver este problema mediante el uso de “plugins” de terceros para imitar el funcionamiento que tenía el sistema web. Por otra parte, mejoré mis conocimientos en CSS (Cascading Style Sheets) ya que tuve que adaptar algún contenido para que se mostrase de una forma más atractiva y consistente con el diseño de todas las páginas. Además de esto, aprendí a configurar un servidor para que pudiera acceder a él cualquier usuario que quisiera acceder al sistema de páginas web migrado.

Por lo que se refiere al Trabajo de Fin de Grado, hemos estudiado basándonos en dos teoremas importantes del Análisis Matemático, las propiedades aproximativas de algunas redes neuronales artificiales. Estas propiedades nos permiten trabajar con funciones más sencillas en lugar de otras más complejas que dificultarían su uso.



# Bibliografía

FRANCISCO JOSÉ PALACIOS, Herramientas en GNU/Linux para estudiantes universitarios, Redes Neuronales con GNU/Linux, [https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes\\_neuronales/curso-glisa-redes\\_neuronales-html/index.html](https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes_neuronales/curso-glisa-redes_neuronales-html/index.html) Sec.1-3 [Consulta: Octubre 2016]

POPESCU M. C., BALAS V. E., PERESCU-POPESCU L., MASTORAKIS N, *Multilayer Perceptron and Neural Networks*, WSEAS Trans Circuits Syst 7: p.579–588, 2009.

LUZ GLORIA TORRES, *Redes Neuronales y Aproximación de Funciones*. Boletín de Matemáticas, [S.l.], v. 1, n. 2, p. 35-58, nov. 1994. ISSN 2357-6529.

H. L. ROYDEN, *Real Analysis* Macmillan Publishing Company, Stanford University (California), p.171-174, ISBN 0-02-404150-5, 1968.

LIANG JIN, MADAN M. GUPTA, y PETER N. NIKIFORUK, *Approximation Capabilities of Feedforward and Recurrent Neural Networks*, en Intelligent Control Systems, M. M. GUPTA y N. K. SINHA, Eds. Piscataway, NJ IEEE Press, 1995.

NEIL E. COTTER, *The Stone-Weierstrass Theorem and Its Application to Neural Network* en IEEE Transactions on Neural Networks, Vol. 1, No. 4, 1990.