

Tecnológico de Costa Rica  
Escuela de Ingeniería Electrónica  
Programa de Maestría en Electrónica



**Identificación de la temperatura electrónica de un plasma frío  
por medio de redes neuronales artificiales**

Tesis de Maestría para optar por el título de  
Magister Scientiae en Ingeniería Electrónica  
Énfasis en Sistemas Empotrados

Jeferson González Gómez

Cartago, 10 de diciembre de 2015



Declaro que la presente tesis ha sido realizada enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas. En consecuencia, asumo la responsabilidad total por la tesis realizada y por su contenido del correspondiente.

Jeferson González Gómez

Cartago, 10 de diciembre de 2015


Céd: 2-0661-0852



Instituto Tecnológico de Costa Rica  
Escuela de Ingeniería Electrónica  
Tesis de Maestría  
Tribunal Evaluador


Tesis defendida ante el presente Tribunal Evaluador como requisito para optar por el título de Magister Scientiae en Ingeniería Electrónica en el Instituto Tecnológico de Costa Rica.

Miembros del Tribunal




---

Dr. Iván Vargas Blanco  
Lector



---

Dr. Juan Luis Crespo Mariño  
Lector



---

MSc. Eduardo Canessa Montero  
Profesor Asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, 9 de diciembre de 2015



# Resumen

En dispositivos de fusión nuclear por plasma, el desempeño de los experimentos está relacionado con la capacidad de sus diagnósticos, esto es, la capacidad de obtener de manera correcta y efectiva los parámetros del plasma involucrados en cada experimento. En el área de control del plasma para dispositivos de confinamiento magnético como Tokamaks o Stellarators, crear, mantener y terminar una descarga de plasma son actividades ligadas a los parámetros mismos del plasma.

Las técnicas convencionales de extracción de parámetros del plasma, como es el caso de la Sonda de Langmuir, implican una medida indirecta y, generalmente, un procesamiento fuera de línea para llegar a un parámetro determinado. Esta naturaleza fuera de línea restringe cualquier tarea de control del plasma en tiempo real.

En esta investigación se pretende abordar el problema de la extracción paramétrica del plasma, específicamente la temperatura electrónica, por medio de técnicas de recuperación rápida en línea, enfocadas hacia las ciencias de la computación y la inteligencia artificial, como lo son las redes neuronales artificiales.

**Palabras clave:** plasma, redes neuronales artificiales, temperatura electrónica, Langmuir





# Abstract

In plasma nuclear fusion devices, the performance of the experiments is related to their diagnostic capability, that is, the capability to effectively obtain the plasma parameters involved in each experiment. In the control area, for plasma magnetic confinement devices as Tokamaks or Stellarators, create, maintain and finish a plasma discharge are tasks linked to plasma parameters themselves.

Conventional techniques for extracting plasma parameters, such as the Langmuir probe, involve an indirect measure and usually require off-line processing to reach a certain parameter. This offline nature restricts any plasma control task in real time.

This thesis aims to address the problem of parametric plasma extraction, specifically the electron temperature, through techniques of rapid online recovery, focused towards computer science and artificial intelligence, as such as artificial neural networks.

**Keywords:** plasma, artificial neural network, electronic density, electronic temperature, Langmuir



*a mis queridos padres*



# Índice general

Índice de figuras	iii
Índice de tablas	v
<b>1 Introducción</b>	<b>1</b>
1.1 Entorno . . . . .	1
1.2 Justificación . . . . .	3
1.3 Problema . . . . .	3
1.4 Enfoque de Solución . . . . .	4
1.5 Objetivos . . . . .	6
1.6 Estructura . . . . .	6
<b>2 Marco teórico</b>	<b>7</b>
2.1 Métodos de medición . . . . .	7
2.1.1 Sonda de Langmuir . . . . .	7
2.1.2 Thomson Scattering . . . . .	8
2.2 Recuperación rápida de parámetros del plasma . . . . .	10
2.2.1 Redes Neuronales Artificiales . . . . .	10
2.2.2 Regresión por redes neuronales . . . . .	12
2.3 Identificación de parámetros del plasma por Redes Neuronales Artificiales .	13
<b>3 Metodología</b>	<b>17</b>
3.1 Investigación bibliográfica . . . . .	17
3.2 Selección de las plataformas . . . . .	18
3.3 Recolección experimental de datos sobre plasmas fríos . . . . .	18
3.4 Definición de topología de la red neuronal . . . . .	19
3.5 Entrenamiento . . . . .	19
3.6 Validación cruzada . . . . .	20
3.7 Implementación final . . . . .	20
<b>4 Identificación de la temperatura electrónica por medio de redes neuronales artificiales</b>	<b>23</b>
4.1 Extracción de datos . . . . .	23
4.1.1 Cálculo de la temperatura electrónica . . . . .	24
4.2 Selección de topologías de red . . . . .	26

---

4.3	Definición de características de red neuronal . . . . .	27
4.3.1	Perceptrón Multicapa . . . . .	27
4.3.2	Función de base radial . . . . .	29
4.4	Implementación de prototipos . . . . .	31
<b>5</b>	<b>Resultados y análisis</b>	<b>35</b>
<b>6</b>	<b>Conclusiones y recomendaciones</b>	<b>43</b>
	<b>Bibliografía</b>	<b>45</b>
<b>A</b>	<b>Datos para determinación de número de capas y neuronas</b>	<b>49</b>
A.1	Perceptron multicapa . . . . .	49
A.2	Función de base radial . . . . .	51
<b>B</b>	<b>Archivos de redes para biblioteca FANN</b>	<b>53</b>
B.1	mlp.net . . . . .	53
B.2	rbf.net . . . . .	54
	<b>Índice alfabético</b>	<b>57</b>

# Índice de figuras

1.1	Tokamak esférico MEDUSA-CR [1]. . . . .	2
1.2	Diagrama computacional del Stellarator SCR-1 [2]. . . . .	2
1.3	Diagrama de bloques general de la solución. . . . .	4
1.4	Diagrama de flujo del proceso de aplicación y evaluación de redes neuronales artificiales. . . . .	5
2.1	Curvas ideales corriente-tensión para el método de la Sonda de Langmuir [3].	8
2.2	Medición de temperatura electrónica por Thomson scattering en el Tokamak T3-A [4]. . . . .	9
2.3	Representación de una neurona artificial simple [5]. . . . .	10
2.4	Funciones de activación comunes: (a) función lineal, (b) función umbral, (c) función umbral lineal y (d) sigmoideal simétrica [5]. . . . .	11
2.5	Topología de red perceptrón multicapa (MLP) [5]. . . . .	12
2.6	Topología de red recurrente tipo Hopfield [6]. . . . .	13
2.7	Topología de red neuronal doble (DANN) [7]. . . . .	14
2.8	Topología de red con función base radial (RBF) [5]. . . . .	15
3.1	Plataforma NI PXIe 8135 [8]. . . . .	21
4.1	Diagrama de flujo del cálculo de la temperatura electrónica según método de Langmuir. . . . .	25
4.2	Logaritmo del error cuadrático medio contra número de iteraciones de entrenamiento para diferentes funciones de activación en red <i>MLP</i> . . . . .	28
4.3	Función de costo para diferentes configuraciones de red <i>MLP</i> . . . . .	30
4.4	Logaritmo del error cuadrático medio contra número de iteraciones de entrenamiento para diferentes funciones de activación en red <i>RBF</i> . . . . .	31
4.5	Función de costo para diferente número de neuronas de capa oculta en red <i>RBF</i> . . . . .	31
4.6	Red neuronal RBF en LabVIEW . . . . .	34
4.7	Red neuronal MLP en LabVIEW . . . . .	34
5.1	Curva típica de corriente-tensión obtenida experimentalmente en sonda de Langmuir. . . . .	36
5.2	Curva de entrenamiento de red MLP. . . . .	38
5.3	Curva de entrenamiento de red RBF. . . . .	39





# Índice de tablas

5.1	Comparación entre $T_e$ obtenida algorítmicamente y referencia externa, con datos de prueba de [9] . . . . .	36
5.2	Comparación entre $T_e$ obtenida algorítmicamente y referencia externa, con datos obtenidos en experimento en <i>e-lab</i> . . . . .	37
5.3	Validación cruzada de red MLP . . . . .	40
5.4	Validación cruzada de red RBF . . . . .	40
5.5	Error cuadrático medio y porcentaje de error absoluto en validación cruzada de las redes MLP y RBF. . . . .	40
5.6	Tiempo de ejecución de las redes neuronales artificiales en diferentes plataformas . . . . .	41
A.1	Función de costo para selección de número de neuronas de capa oculta en red neuronal MLP. . . . .	49
A.2	Función de costo para selección de número de neuronas de capa oculta en red neuronal RBF. . . . .	51



# Capítulo 1

## Introducción

### 1.1 Entorno

El Laboratorio de Plasmas para la Energía de Fusión y Aplicaciones del Instituto Tecnológico de Costa Rica (PlasmaTEC), pretende formar recurso humano para desarrollar las tecnologías de plasma en el país, en las áreas de fusión nuclear por plasma y aplicaciones industriales.

Desde el punto de vista de la fusión por plasma, el laboratorio se encuentra trabajando en dos tipos de dispositivo de confinamiento magnético: MEDUSA-CR (tipo Tokamak) y SCR-1 (tipo Stellarator). Un *Tokamak*, Fig.1.1, es un dispositivo de confinamiento de plasma construido en forma de toroide, en el que el plasma es confinado por medios magnéticos [10], basado en corrientes inducidas dentro del plasma. Un *Stellarator* como el mostrado en la Fig.1.2 posee una asimetría toroidal, por medio de la cual mantiene una configuración magnética de un campo externo, producido por una serie de bobinas modulares, sin necesidad de tener un campo magnético interno producto de corrientes en el plasma [11, 12].

En los dispositivos de fusión nuclear por plasma, los diagnósticos, o métodos de extracción de parámetros del plasma, como la temperatura electrónica, tienen un papel fundamental en los experimentos, ya que permiten caracterizar el plasma producido en cada descarga. En el Laboratorio de Plasma para la Energía de Fusión y Aplicaciones, existe actualmente un diagnóstico conocido como Sonda de Langmuir (ver 2.1.1), mediante el cual, tras un procesamiento fuera de línea, se obtienen la temperatura y densidad electrónica para condiciones de experimento determinadas.

Otra de las áreas de trabajo del laboratorio es en *plasmas fríos*, cuya temperatura electrónica es menor que  $10^4 K$  [13]. Esta área es la que actualmente posee más desarrollo en aplicaciones industriales y medicas como deposición y tratamiento de materiales, nitruración por plasma, esterilización de equipo, entre otras. Es por esto que, para objetivos de esta tesis, se utilizarán plasmas fríos, ya que es de interés para las aplicaciones mencionadas anteriormente conocer y controlar los parámetros de temperatura y densidad electrónica,

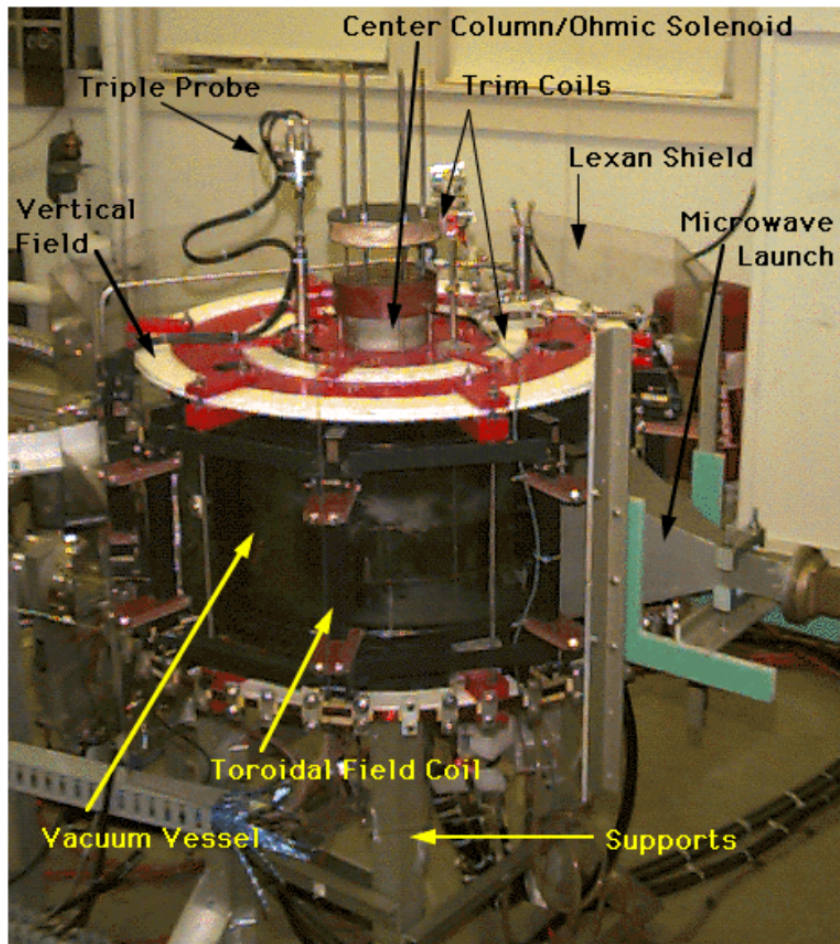


Figura 1.1: Tokamak esférico MEDUSA-CR [1].

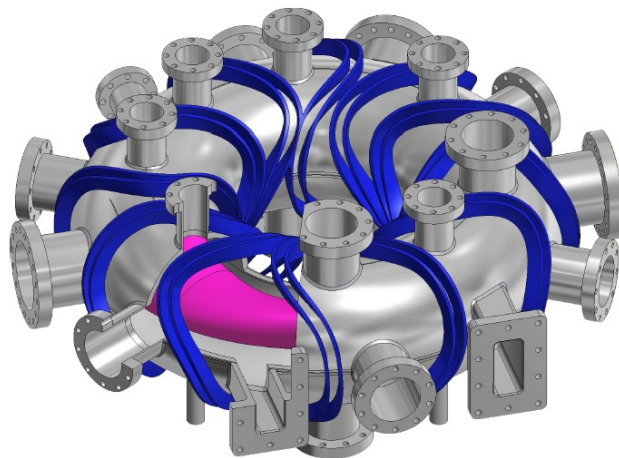


Figura 1.2: Diagrama computacional del Stellarator SCR-1 [2].

propios de cada aplicación.

## 1.2 Justificación

En los dispositivos de fusión nuclear por plasma, se ha establecido [14] que el desempeño de los experimentos está directamente relacionado con la capacidad de sus diagnósticos, esto es, la capacidad de obtener de manera correcta y efectiva los parámetros del plasma que describen el comportamiento del mismo. El perfil de temperatura electrónica, en el caso del experimento ITER por ejemplo, es el mayor indicador del desempeño del plasma [14].

En el área de control del plasma para dispositivos de confinamiento magnético como Tokamaks o Stellarators, crear, mantener y terminar una descarga de plasma son actividades ligadas a los parámetros mismos del plasma [7]. Los procedimientos de control incluyen como primer paso contar con un medio para la medición de los parámetros por métodos convencionales o diagnósticos como Sonda de Langmuir o Thomson Scattering. El siguiente paso involucra el cálculo propio de los parámetros por medio de algoritmos computacionales. Por último, debe existir una manera de calcular la respuesta del sistema, así como de modificar los parámetros del plasma, basado en un estado del plasma deseado como referencia [7]. Este proceso requiere conocer el estado actual del plasma en cualquier instante de tiempo, por lo que para sistemas de confinamiento magnético, la tarea de control del plasma debe ser realizada en tiempo real, y la identificación debe ser capaz de realizarse rápidamente en cada instante de tiempo.

La característica fuera de línea del proceso actual de extracción de la temperatura y densidad del plasma limita significativamente su aplicación en experimentos de control del plasma en tiempo real, por ejemplo control de posición, perfiles específicos de temperatura y densidad, así como estabilidad y dispersión del plasma.

Así mismo, debido a la naturaleza compleja, no-lineal, del plasma, no es posible obtener una solución analítica del sistema [15], por ejemplo un modelado matemático en función de entrada y salidas. Esta naturaleza descarta además técnicas de identificación comúnmente utilizadas como la regresión lineal. Las soluciones por aproximaciones numéricas, que se consideran la técnica estándar para este tipo de problemas, implican posprocesamiento, lo que no es aceptable en sistemas de control de tiempo real. Métodos de identificación o recuperación rápida de los parámetros, como es el caso de las redes neuronales artificiales, son entonces indispensables y óptimos en este tipo de sistemas [7, 15].

## 1.3 Problema

Los métodos tradicionales de extracción de los parámetros del plasma brindan en general una medición indirecta y requieren un procesamiento fuera de línea, lo que limita cualquier tarea de control en tiempo real sobre los distintos experimentos. Dado lo anterior:

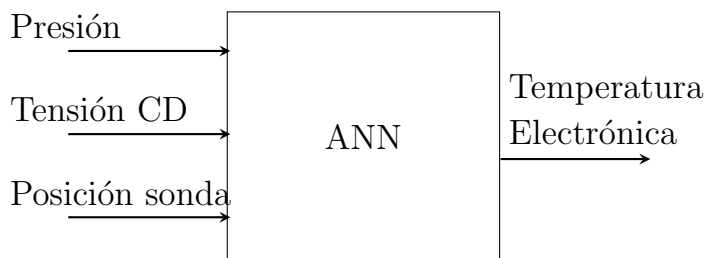
¿Es posible obtener la temperatura electrónica del plasma por un medio de recuperación rápida como las redes neuronales artificiales?

## 1.4 Enfoque de Solución

Con el fin de responder a la pregunta anterior, se planteará una estrategia de identificación paramétrica de la temperatura del plasma para un plasma frío del tipo CD (generado por tensión en corriente directa), por medio de redes neuronales artificiales, para demostrar la validez de dicha técnica como mecanismo de recuperación paramétrica, rápida y en tiempo real. Se utilizarán como entradas las variables de presión en la cámara de vacío (mTorr), posición de la sonda de Langmuir (mm) y tensión eléctrica en corriente directa (CD) aplicada al plasma (V).

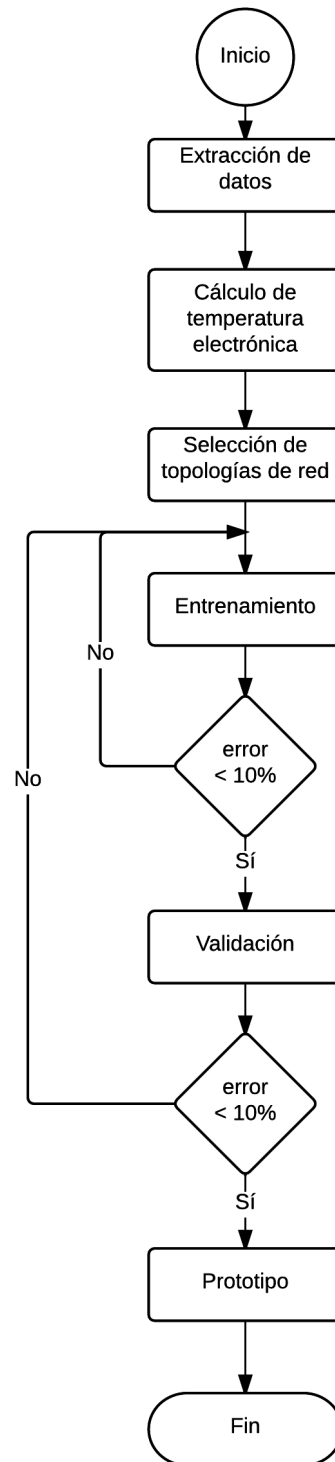
Se deberá seleccionar una arquitectura de red neuronal adecuada, de forma que el entrenamiento de la red y la validación cruzada, sobre los datos obtenidos por medio de la sonda de Langmuir, permitan extraer la temperatura del plasma, de una manera correcta, con un error promedio menor al 10%.

En la Fig.1.3 se muestra un diagrama de bloques de la solución planteada. Dependiendo de la arquitectura de la red seleccionada, se implementará la misma en una aplicación descrita en lenguaje C, utilizando bibliotecas especializadas para tal fin, como la biblioteca *Fast Artificial Neural Network* (FANN)[16]. Desde el punto de vista de implementación, esta biblioteca permite exportar los parámetros de la red neuronal artificial en un archivo de texto (.net), para de esta forma, utilizarla en cualquier otra plataforma de hardware y software, lo que además favorece su escalabilidad.



**Figura 1.3:** Diagrama de bloques general de la solución.

La aplicación y evaluación del método de redes neuronales se muestra en el Fig.1.4. El proceso (ver sección 3) inicia con el paso de obtención experimental del conjunto de datos de entrenamiento, posteriormente, una vez determinada la arquitectura de red, se deberá realizar el entrenamiento de la red neuronal artificial hasta que el error medio cuadrático se encuentre dentro del rango considerado como aceptable para la aplicación. Para el caso de esta investigación, el error en el entrenamiento considerado como aceptable será menor o igual a 10%. Posteriormente se validará la red ya entrenada contra un conjunto de datos diferente del que fue entrenado, proceso que se conoce como *validación cruzada*. Finalmente, se realizará un prototipo para evaluar el desempeño del sistema en términos de cumplimiento de tiempos de ejecución en tiempo real.



**Figura 1.4:** Diagrama de flujo del proceso de aplicación y evaluación de redes neuronales artificiales.

## 1.5 Objetivos

En esta sección se describe el objetivo general y los objetivos específicos a cubrir en el desarrollo de esta tesis, con el fin brindar un panorama más claro de los alcances y límites de la misma.

### Objetivo General

Evaluar el uso de redes neuronales artificiales como método de identificación de la temperatura electrónica en un plasma frío para dispositivos de generación y confinamiento de plasmas.

### Objetivos Específicos

- Determinar una topología de red neuronal apropiada para la determinación paramétrica del plasma.
- Aplicar un entrenamiento a la red neuronal seleccionada para un rango de error aceptable.
- Validar los resultados de la red neuronal contra la teoría clásica de Sonda de Langmuir.
- Evaluar una plataforma de hardware para la implementación de la red neuronal conforme a requerimientos mínimos de tiempo real y procesamiento paralelo.

## 1.6 Estructura

En este documento se presenta el proceso de evaluación de las redes neuronales como método de identificación paramétrica para un plasma frío del tipo CD. En el siguiente capítulo, a modo de marco teórico, se presentan los conceptos teóricos y estado del arte en técnicas de identificación de parámetros, haciendo énfasis en las redes neuronales artificiales. El capítulo 3 describe el marco metodológico utilizado como referencia para el desarrollo de la tesis. La descripción de la solución, así como las decisiones de diseño e implementación se describen en el capítulo 4. El capítulo 5, presentan los principales resultados obtenidos y su respectivo análisis. Finalmente, las conclusiones de todo el desarrollo planteado anteriormente, se describen en el capítulo 6.



# Capítulo 2

## Marco teórico

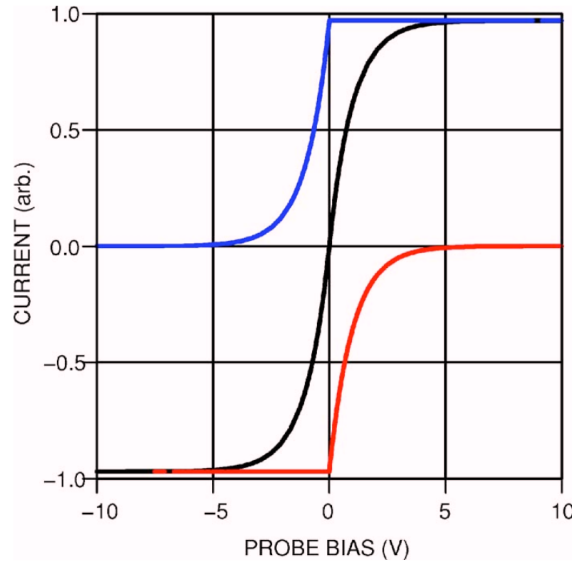
Esta sección pretende describir, a modo de revisión, los principales métodos para la medición e identificación de los parámetros del plasma. Los métodos tradicionales para la medición de los parámetros (temperatura electrónica y densidad) se presentan en la Sec. 2.1. En la Sec. 2.2 se describe la metodología de recuperación rápida de los parámetros, haciendo énfasis en la técnica de redes neuronales artificiales.

### 2.1 Métodos de medición

#### 2.1.1 Sonda de Langmuir

En el estudio de la física de plasmas, se utiliza la técnica de *sonda de Langmuir*, propuesta en 1926 [17], para medir la densidad del plasma, la temperatura electrónica y el potencial del plasma. Una sonda de Langmuir consiste en un cable, electrodo o disco de metal que se inserta en el plasma y se polariza eléctricamente respecto a nodo de referencia, con el fin de recolectar corrientes de iones o electrones [3]. El método de Langmuir consiste en obtener curvas de corriente vs. tensión (I-V) características de la sonda, como las mostradas en la Fig.2.1, mientras se aplica una tensión de polarización  $V_B$  en un barrido desde potenciales negativos hasta positivos [17, 3]. En la Fig.2.1, la curva negra representa la corriente total en la sonda, la azul la corriente iónica y en rojo la corriente electrónica. La tensión de polarización en la que la corriente total en la sonda cambia de signo, de negativo a positivo, se conoce como *potencial flotante* del plasma ( $V_f$ ).

El *potencial del plasma* ( $V_p$ ) es el potencial en la posición específica de la sonda. Para valores de  $V_B \gg V_p$ , la sonda recolecta la corriente de saturación electrónica  $I_{es}$  y para valores de  $V_B \ll V_p$ , todos los electrones en la sección de la sonda son repelidos, de forma que  $I_{es} = 0$ . La corriente electrónica está, por lo tanto, relacionada con la tensión de polarización. Esta relación [3] viene dada por medio de:



**Figura 2.1:** Curvas ideales corriente-tensión para el método de la Sonda de Langmuir [3].

$$I_e(V_B) = \begin{cases} I_{es} \exp[-e(V_p - V_B)/kT_e], & V_B \leq V_p \\ I_{es}, & V_B > V_p. \end{cases} \quad (2.1)$$

El valor máximo de la curva de la primera derivada de la corriente en la sonda, en la Ec.2.1, con respecto a la tensión aplicada, ocurre en  $V_p$  [18]. Luego, la relación entre la corriente y la primera derivada de la corriente en  $V_p$  [18] es :

$$\left. \frac{I_e(V_B)}{dI(V_B)/dV_B} \right|_{V_B=V_p} = \frac{kT_e}{e}, \quad (2.2)$$

esto es, la temperatura electrónica del plasma, en eV.

Finalmente, para el caso en que la temperatura electrónica es igual a la temperatura iónica, existe una relación entre la densidad electrónica ( $n_e$ ), la temperatura electrónica ( $T_e$ ) y la corriente de saturación iónica ( $I_{es}$ ), que viene dada por:

$$I_{is} = \frac{1}{4} e n_e \sqrt{8kT_e/\pi m_i} A_{sonda}, \quad (2.3)$$

donde  $e$  representa la carga del electrón,  $m_i$  es la masa del ion y  $A_{sonda}$  corresponde al área de la sonda. Posterior al barrido, mediante las ecuaciones 2.1, 2.2, 2.3, se pueden obtener los valores de temperatura y densidad electrónica, mediante posprocesamiento de los datos recogidos durante el experimento.

### 2.1.2 Thomson Scattering

La técnica de *Thomson scattering* es otro método diagnóstico bastante conocido para la medición de los parámetros básicos del plasma [19] en dispositivos de fusión por plas-

ma, como Tokamaks. Las mediciones de Thomson scattering, en la práctica, permiten determinar la temperatura electrónica y la densidad electrónica en plasmas en equilibrio.

El método de Thomson scattering consiste en hacer incidir una onda electromagnética en el plasma que, por medio el proceso de esparcimiento (*scattering*), hace que electrones libres sean acelerados e irradian. Esta radiación producida en los electrones libres tiene la misma frecuencia que la radiación incidente [4].

El factor de forma de la onda dispersa (*scattered*) viene dado por:

$$S(\mathbf{k}, \omega) = \frac{n_e}{k} \left( \frac{m}{2\pi k_B T_e} \right)^{\frac{1}{2}} \exp \left[ \frac{-m\omega^2}{2k^2 k_B T_e} \right] \quad (2.4)$$

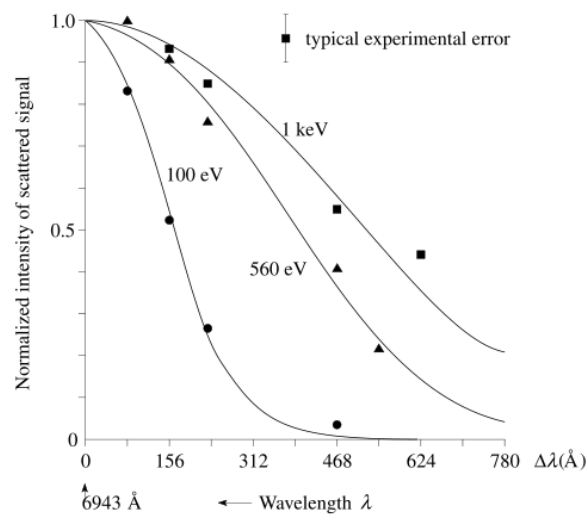
Donde  $k$  es el número de onda,  $\omega$  es la frecuencia y  $k_B$  corresponde a la constante Boltzmann.

La magnitud máxima del espectro de Fourier del *factor de forma*  $S(\mathbf{k}, \omega)$  de la onda dispersa determina la densidad electrónica del plasma, mientras que la mitad del ancho del valor máximo del espectro  $\Delta\omega$ , que viene dado por:

$$\Delta\omega = 4\omega_0 \left( \frac{2K_B T_e \ln 2}{mc^2} \right)^{\frac{1}{2}} \sin \frac{\theta}{2} \quad (2.5)$$

da un medida directa de la temperatura electrónica del plasma[4].

El perfil de temperatura de la Fig.2.2, obtenido de la primera aplicación del Thomson scatteringen [20] para el Tokamak T3-A, permitió validar el método como diagnóstico del plasma.



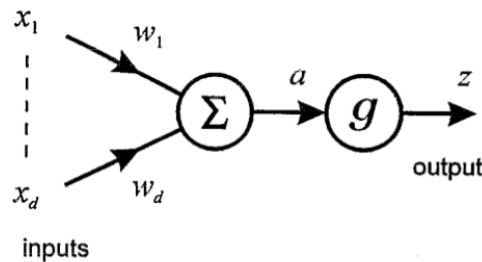
**Figura 2.2:** Medición de temperatura electrónica por Thomson scattering en el Tokamak T3-A [4].

## 2.2 Recuperación rápida de parámetros del plasma

En términos de identificación y control de plasmas, las configuraciones particulares son descritas como soluciones a ecuaciones diferenciales no-lineales de las que, debido a su naturaleza no-lineal, no es posible obtener una solución analítica [15], esta naturaleza descarta además técnicas de identificación comúnmente utilizadas como la regresión lineal. Las soluciones por aproximaciones numéricas, que se consideran la técnica estándar para este tipo de problemas, implican posprocesamiento, lo que no es aceptable en sistemas de control de tiempo real. Métodos de identificación o recuperación rápida de los parámetros, como es el caso de las redes neuronales artificiales, son entonces indispensables y óptimos en este tipo de sistemas [7, 15], dada su capacidad para adecuarse a sistemas no lineales, sin necesariamente conocer el modelo analítico de los mismos.

### 2.2.1 Redes Neuronales Artificiales

La técnica de redes neuronales artificiales (ANN) nace como un modelo artificial de las capacidades de procesamiento paralelo del cerebro humano [7]. Las redes neuronales artificiales proveen un gran rango de técnicas para la solución de problemas en el reconocimiento de patrones, análisis de datos y control. Una unidad en la red neuronal artificial es llamada *perceptrón* (modelo matemático simplificado de una neurona) y puede ser conceptualizado como una función no-lineal que transforma un conjunto de variables de entrada  $x_i$  en una variable de salida  $z$  [5].



**Figura 2.3:** Representación de una neurona artificial simple [5].

En una neurona simple, como la de la Fig.2.3, la señal de entrada  $x_i$  es multiplicada por un parámetro  $w_i$ , conocido como *peso*, y luego se suma a todas las demás entradas *ajustadas* (por sus pesos correspondientes) para dar el resultado de la unidad, de la forma:

$$a = \sum_{i=1}^N w_i x_i + b, \quad (2.6)$$

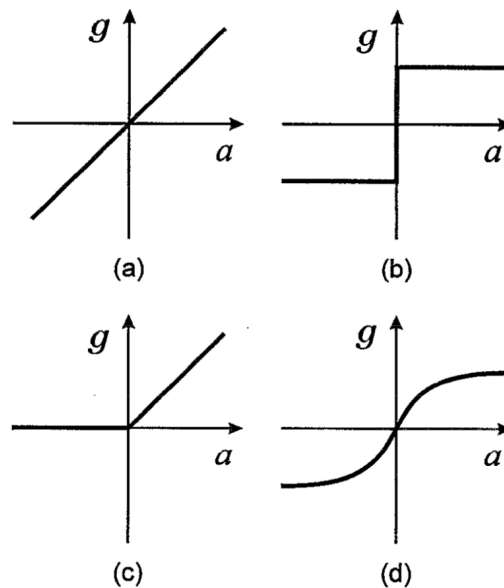
donde el parámetro  $b$ , se denomina *bias* y corresponde de manera análoga al umbral de activación de una neurona biológica. La salida de la unidad  $a$  es luego operada por medio de una *función de activación*  $g(a)$ , cuyo comportamiento suele ser no-lineal y diferenciable.

De esta forma:

$$z = g(a) \quad (2.7)$$

Funciones de activación comunes suelen tener comportamientos sigmoidales. Una función de activación *sigmoidal*, como la de la ecuación 2.8, mapea su salida en el rango  $[0, 1]$  [21]. Otra función de activación comúnmente utilizada es la función *tanh*, que posee un comportamiento similar a la función sigmoidal en el rango  $[0, \infty]$ , pero además mapea su salida al rango  $[-1, 0]$ , para valores de entrada de  $[-\infty, 0]$ . La función de activación  $\tanh(a)$ , o *sigmoidal simétrica*, da lugar a convergencias más rápidas [21]. En la Fig.2.4 se muestran algunas funciones de activación usadas comúnmente.

$$g(a) = \frac{1}{1 + \exp(-a)} \quad (2.8)$$



**Figura 2.4:** Funciones de activación comunes: (a) función lineal, (b) función umbral, (c) función umbral lineal y (d) sigmoidal simétrica [5].

Una red neuronal está compuesta por al menos dos capas. Cada capa corresponde a la unión de un número  $N$  de neuronas simples. El proceso mediante el cual se ajustan los parámetros de peso  $w_i$  de cada neurona en cada capa, se denomina *aprendizaje* o *entrenamiento*. Una de las técnicas más comunes de entrenamiento para redes multicapa con función de activación diferenciable se denomina *retropropagación*. Bajo este método, el error entre la salida y su valor teórico correspondiente se propaga hacia atrás desde la capa de salida a través de la red neuronal [21].

Una vez que se han determinado los pesos, nuevas entradas pueden ser procesadas por la red de una manera rápida [5]. La validación de la red, corresponde el proceso evaluador de las salidas de la red neuronal con respecto a un valor esperado. La técnica de *validación cruzada* es ampliamente utilizada en este contexto. La idea de esta técnica es separar la

base de datos en un conjunto para entrenamiento y otro para validación, de esta manera ampliar la capacidad de la red para generalizar [22]. La validación y entrenamiento, por lo tanto, se realizan sobre conjuntos diferentes.

### 2.2.2 Regresión por redes neuronales

En el ámbito del reconocimiento de patrones, la tarea de regresión consiste en representar, mediante salidas, valores de variables continuas. La regresión puede ser vista como un caso particular de aproximación de funciones [21] y/o parámetros. En este ámbito, la importancia de las redes neuronales, como método de estimación paramétrica, es que ofrecen un marco poderoso y general para mapeos de funciones no-lineales (comportamientos) de múltiples señales de entrada hacia múltiples señales de salida [21]. Desde un punto de vista de teoría de sistemas, la técnica de redes neuronales brinda un mapeo no-lineal para sistemas tanto estáticos como dinámicos, y permite la identificación en línea de estos sistemas. De hecho, la habilidad de las redes neuronales para aproximar funciones no-lineales con suficiente precisión las hacen el principal candidato para la representación de plantas no-lineales en sistemas dinámicos. [6]

En [6] se presentan ejemplos de un acercamiento *unificado* de redes neuronales, combinando topologías como el perceptrón multicapa (*MLP*), Fig.2.5, y las redes de retroalimentadas o recurrentes (*RNN*), Fig.2.6 para la identificación y control de sistemas dinámicos, en los que la falta de un modelo analítico y la complejidad intrínseca del sistema, hacen que el enfoque basado en redes neuronales artificiales genere resultados satisfactorios.

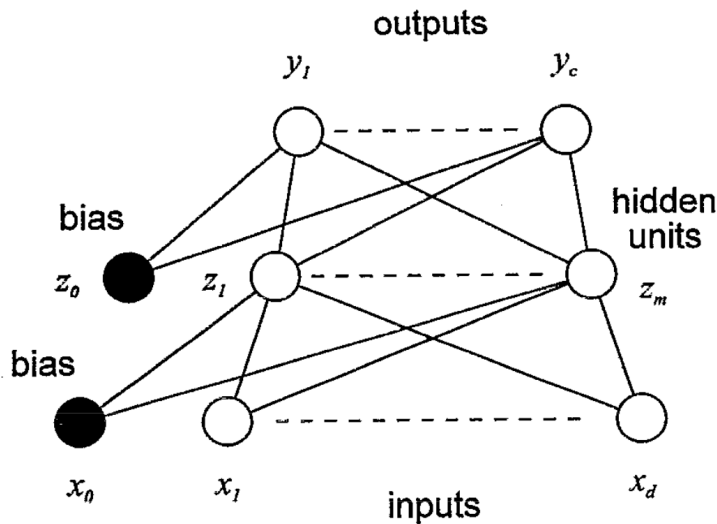


Figura 2.5: Topología de red perceptrón multicapa (MLP) [5].

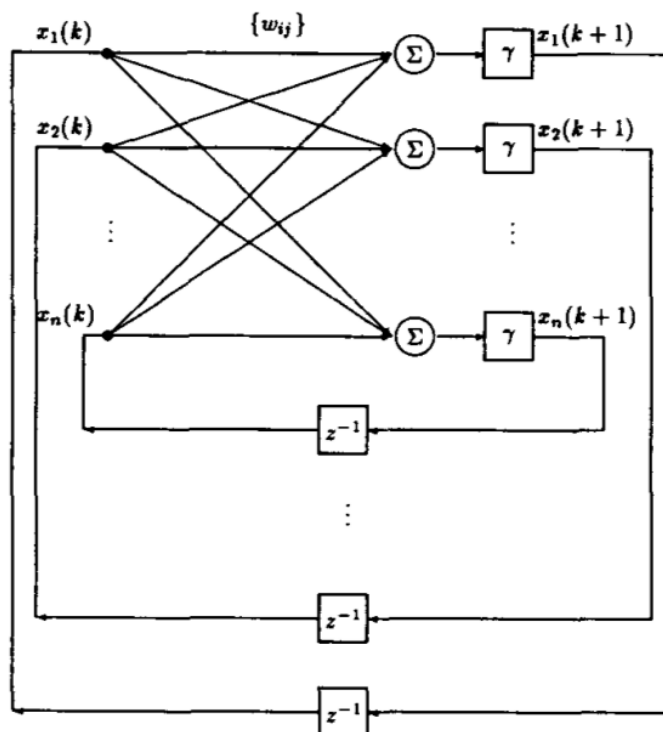


Figura 2.6: Topología de red recurrente tipo Hopfield [6].

## 2.3 Identificación de parámetros del plasma por Redes Neuronales Artificiales

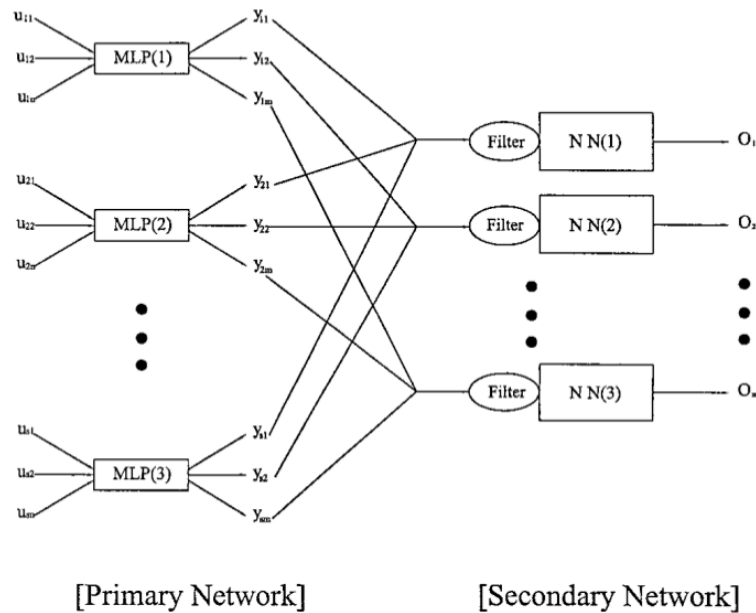
La técnica de redes neuronales artificiales ha sido utilizada previamente como método de identificación de parámetros del plasma, especialmente aquellos que son requeridos en procesos de estabilidad, equilibrio y control de posición del mismo.

En [23] se presenta uno de los primeros acercamientos a extracción rápida de parámetros del plasma por medio de redes neuronales artificiales. En este caso, se desarrolla una topología de perceptrón multicapa (función de activación sigmoideal para capa oculta, lineal para capa de salida) para la extracción de 13 parámetros geométricos del plasma en el Tokamak DIII-D, con un entrenamiento de 500 muestras de puntos de equilibrio del plasma. Para cada set de entrada, la red neuronal MLP presentó mejor desempeño que el ajuste lineal (método previo).

El primer uso de redes neuronales para el control propiamente del plasma para un dispositivo Tokamak se documenta en [15]. En este escenario, nuevamente se implementa un red neuronal perceptrón multicapa para la identificación y control de parámetros de equilibrio del plasma (elongación, triangularidad, distancia vertical del plasma con respecto al centro y distancia radial del plasma con respecto al centro) en el Tokamak COMPASS. Para el parámetro de la elongación, por ejemplo, el error residual típico logrado se encontró en el rango de 0.07 a 1.15, con las limitaciones de implementación de red en Hardware. La

implementación presentó una función de activación  $\tanh$  y un total de 4 neuronas de capa oculta.

Nuevamente en el área de equilibrio de plasma, soluciones más recientes han implementado otras topologías de redes para la identificación paramétrica. Segupta y Rajan [7] realizan una comparación entre técnicas de redes neuronales modificadas para la recuperación paramétrica rápida, con el fin de control en tiempo real de la estabilidad del plasma. Entre las técnicas descritas destacan la aplicación de redes neuronales dobles (*DANN*), Fig. 2.7, redes basadas en pre-procesamiento y reducción dimensional por análisis de componentes principales (*PCANN*) y redes basadas en regresión polinomial fraccionaria (FP). La identificación, basada en las diferentes topologías, tuvo un total de 72 entradas con 23 salidas (correspondientes a parámetros geométricos) y una función de activación sigmooidal simétrica. Para este caso la combinación de perceptrón multicapa, junto con pre-procesamiento por PCA presentó los menores errores en todos los parámetros, así como el mejor desempeño en la identificación.

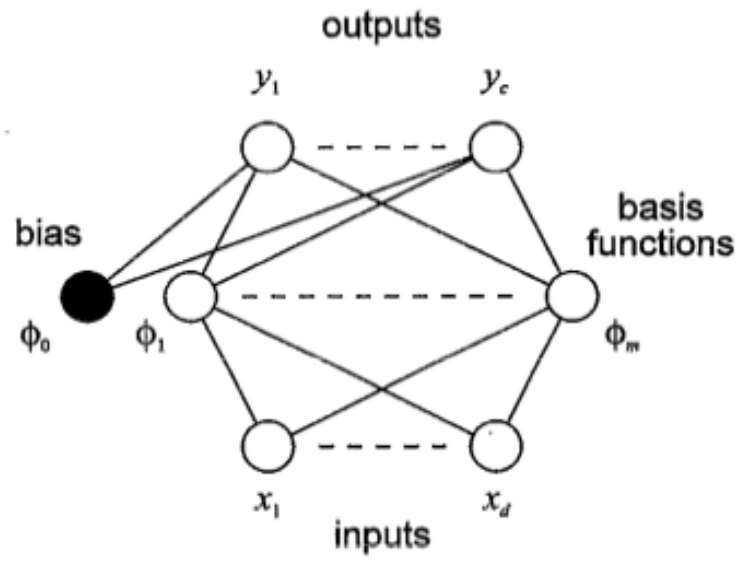


**Figura 2.7:** Topología de red neuronal doble (DANN) [7].

En el caso de plasmas de baja temperatura (plasmas fríos) las técnicas de identificación se enfocan en los parámetros de densidad y temperatura electrónica. En esta área, en [24] se realiza una caracterización de la densidad del plasma para dos modelos de redes neuronales: uno basado en perceptrón multicapa y el otro basado en red con función base radial (*RBF*), como la mostrada en la Fig 2.8. El experimento desarrollado contempla 5 variables de entrada (posición radial, posición axial, potencia de microonda, presión del gas en la cámara y corriente en las bobinas) y una única salida: la densidad electrónica. Los datos de entrenamiento corresponden a 300 datos seleccionados aleatoriamente, obtenidos por medio de la técnica de la Sonda de Langmuir. Para ambas implementaciones de red neuronal artificial, el error entre los datos medidos por medio de la sonda y los



determinados por las redes neuronales se mantuvo menor al 7%.



**Figura 2.8:** Topología de red con función base radial (RBF) [5].



# Capítulo 3

## Metodología

En este capítulo se describen los aspectos metodológicos necesarios para llevar a cabo la solución al problema de la identificación rápida de la temperatura electrónica del plasma por medio de redes neuronales artificiales. La metodología puede clasificarse de la siguiente manera: en la sección 3.1 se describe el proceso de investigación sobre el tema. El proceso de selección de plataforma se presenta en la sección 3.2. En la sección 3.3 se desarrollan los aspectos metodológicos que rigen los experimentos para la obtención de los datos. La definición de la topología correcta, su entrenamiento y validación se presentan en la secciones 3.4, 3.5 y 3.6 respectivamente. Por último, los detalles de implementación final se describen en la sección 3.7.

### 3.1 Investigación bibliográfica

Como primer paso del marco metodológico, se realizó una búsqueda bibliográfica sobre los principales aspectos relacionados con la identificación paramétrica. En el capítulo 2 se presenta el resultado del proceso de investigación. Para llevar a cabo este paso, se realizó búsquedas en las bases de datos adquiridas por la institución como: EBSCO, Springer, Science Direct, IEEE Xplore, KNovel, entre otras, así como otros medios digitales de información como Google Scholar, utilizando además gestores bibliográficos como es el caso de Zotero y RefWorks, para un manejo adecuado de las referencias. Los temas abordados en la investigación involucran, en primera instancia, métodos tradicionales de adquisición de los parámetros del plasma como la Sonda de Langmuir y *Thomsom Scattering*. Se abarcó, adicionalmente, el tema de redes neuronales artificiales, desde sus fundamentos hasta aspectos más avanzados de las mismas. Un punto importante fue la investigación sobre posibles implementaciones en C, como es el caso de la biblioteca *FANN*. Finalmente se abarcó aspectos del estado del arte en identificación paramétrica utilizando redes neuronales artificiales como principal método.

La investigación realizada permite sustentar teóricamente la solución al problema planteado, ya que brinda las bases para la aplicación del método de redes neuronales artificiales

para la identificación rápida de la temperatura del plasma.

## 3.2 Selección de las plataformas

Desde el punto de vista de la solución como un todo, se deberán seleccionar y utilizar diferentes plataformas. El proceso de adquisición de datos (sección 3.3) sobre los diferentes experimentos se desarrollará en una plataforma NI PXIe, ya que cuenta con los diferentes módulos de adquisición y comunicación con los diferentes dispositivos involucrados. La programación, por lo tanto, será llevada a cabo mediante la plataforma NI LabVIEW. Por otra parte, por la facilidad, documentación y libre acceso, se podrán utilizar bibliotecas sobre redes neuronales artificiales, como es el caso de la *FANN*, con las que, a modo de prototipado, se podrá realizar el entrenamiento de la red neuronal artificial. Esta biblioteca, por ejemplo, está descrita en lenguaje C, lo que permite que su implementación en diferentes plataformas sea sencilla.

Desde el punto de vista de implementación, la evaluación de la plataforma debe considerar que el número de capas, así como de neuronas, afecta en la cantidad de procesamiento y memoria que se requiere en el mismo. El tiempo límite para realizar la tarea de identificación (*deadline*) deberá ser determinístico, lo que implica que el sistema como tal deberá situarse en un ambiente de tiempo real. Un requisito importante para el sistema debe ser, por lo tanto, el tiempo real, es por esto que la topología debe ser capaz de identificar los parámetros con una precisión, en el rango establecido previamente, con la cantidad de tiempo invertido en procesamiento y preprocesamiento tal que el *deadline* no se sobrepase. Técnicas como la computación paralela serán fundamentales en este esquema, de forma que la plataforma deberá soportar a su vez algún grado de paralelismo. Este balance permitirá la evaluación de la plataforma sobre la cual implementar.

## 3.3 Recolección experimental de datos sobre plasmas fríos

Como base para el entrenamiento y validación posterior de la red neuronal, se debe obtener primero los datos experimentales, esto es, un conjunto de datos de forma de pares entrada-salida. Para esto, se deben generar distintos *plasmas fríos* con diferentes condiciones de entrada, para posteriormente aplicar el método de la Sonda de Langmuir, como se definió en 2.1.1. Cada uno de los experimentos debe realizar un barrido de tensión de polarización de la sonda dentro del plasma y obtener el perfil de corriente correspondiente para la misma, en un proceso que toma aproximadamente 5 minutos por descarga de plasma. La aplicación que realiza la adquisición de estos datos se basó en un modificación al trabajo previo realizado por Soto [25], que adquiere hasta un máximo de 2500 puntos de tensión-corriente en el plasma y los almacena en un archivo.

Una vez obtenidas las curvas de tensión-corriente, se debe calcular la temperatura electrónica, como se describió previamente, para cada uno de los experimentos. De esta manera se conforman los pares entrada (presión, posición, tension CD) - salida (temperatura electrónica).

### 3.4 Definición de topología de la red neuronal

La definición de la arquitectura o topología de la red a utilizar es uno de los pasos más importantes en el desarrollo de la solución. De acuerdo con las fuentes bibliográficas consultadas, mencionadas en el capítulo 2, existen diferentes topologías que han demostrado un buen desempeño en materia de identificación paramétrica. Las topologías de perceptón multicapa (MLP) y redes de función de base radial (RBF) han sobresalido en este sentido, por lo que serán las consideradas en la definición.

Un factor importante a considerar en la elección de la topología, es además, el número de neuronas totales en la red, así como la cantidad de capas. En este sentido, el proceso de entrenamiento proveerá información útil al respecto, ya que determinará el desempeño de la red para cada configuración a entrenar. Por lo tanto, este proceso es el que permitirá definir la mejor arquitectura para el problema planteado.

Finalmente se deberá considerar la necesidad de un pre-procesamiento. Como se mencionó previamente, la técnica de análisis de componentes principales (PCA) puede ser una útil para reducir la dimensionalidad del conjunto de entrada, con la desventaja de que aumenta la complejidad del sistema.

### 3.5 Entrenamiento

El entrenamiento de la red neuronal artificial es el proceso mediante el cual cada neurona ajusta su parámetro de peso según una configuración de entrada-salida y un algoritmo determinado. Dichos algoritmos de entrenamiento, como es el caso de la *retropropagación* [21], pueden tomar una considerable cantidad de tiempo dependiendo de la complejidad de la red y el error medio cuadrático deseado. Se define de manera preliminar que el error deseado en el proceso de entrenamiento deberá ser menor al 10%, con el fin de que dichos errores no afecten negativamente el desempeño de la red en la identificación paramétrica.

El entrenamiento permitirá obtener información útil sobre el problema como tal. Un entrenamiento en el que el error no disminuya conforme a la cantidad de iteraciones podría sugerir un cambio en la arquitectura de la red. Además, si el conjunto de datos de entrada es grande, es posible que el entrenamiento requiera un pre-procesamiento o reducción dimensional. Un aspecto relevante es que el entrenamiento se realiza fuera de línea, lo que implica que para efectos de implementación, los recursos computacionales y algorítmicos tienen mayor flexibilidad que la arquitectura de la red como tal.

## 3.6 Validación cruzada

Una vez entrada la red, se deberá verificar el funcionamiento de la misma. Para tal fin se eligió un método de *validación cruzada*, en donde los datos de validación deben ser distintos a los de entrenamiento, así mismo, lo menos similares posible entre sí (pares entrada-salida redundantes). De esta manera, separando datos de entrenamiento de datos de validación se asegura un mejor funcionamiento de la red ante variaciones de las entradas [22].

Para el caso de este trabajo se segmentará el conjunto de datos de entrada en una proporción 90-10, en la que un 90% de los datos generales serán para entrenamiento y un 10% para la validación. En este escenario se considerará como aceptable un porcentaje de error menor al 10%, valor aceptable según opinión de experto e investigadores y que se encuentra dentro del mismo rango que el trabajo realizado en [24].

## 3.7 Implementación final

Posterior al proceso de verificación cruzada de la red neuronal, se procederá a realizar la implementación de la misma. Para esto, se seleccionaron tres plataformas; la primera plataforma seleccionada para la evaluación de las redes corresponde a un computador de uso general Dell Optiplex 7010, bajo el sistema operativo Ubuntu 12.04 en 64 bits. La plataforma posee un procesador Intel I7 (*Quad-core*), de tercera generación, a 3.1GHz. La selección de esta plataforma se basó en contemplarla como referencia, dado su uso genérico y su alto desempeño en términos de frecuencia de operación y número de procesadores.

La segunda plataforma corresponde al dispositivo NI PXIe 8135, en el que se han desarrollado las diferentes aplicaciones concernientes al laboratorio de energía de fusión y aplicaciones del Instituto Tecnológico de Costa Rica. Esta plataforma posee soporte para tiempo real y programación paralela, por medio de los módulos de NI LabVIEW de tiempo real.

Finalmente, se realiza una implementación de las redes neuronales en la plataforma BeagleBone Black. Dicho dispositivo, que posee un procesador AM335X ARM Cortex-A8 a 1GHZ, se utiliza comúnmente como evaluación de prototipos en sistemas embebidos. Dadas las especificaciones de tiempo real de la aplicación, se implementó la misma bajo un sistema operativo GNU Linux, construido a la medida mediante la herramienta del proyecto Yocto [26] con soporte de kernel para tiempo real.

Desde el punto de vista de tiempo real, se establecerá un *deadline* de 1ms para la ejecución de la red neuronal. Este parámetro ha sido seleccionado de forma que para el peor de los casos de tiempos de descarga de plasma, correspondiente a 15ms en MEDUSA-CR, se puedan extraer al menos 15 puntos de temperatura electrónica a lo largo del experimento.

Para el caso de la plataforma NI PXIe 8135, al ser el dispositivo central de control y

adquisición del laboratorio, se desarrollarán los prototipos de red neuronal empleando el entorno de LabVIEW de tiempo real, donde se encuentra actualmente el sistema de adquisición de datos final, así como la rutina de generación de experimentos por medio de la Sonda de Langmuir. La plataforma NI PXIe (Fig.3.1) posee un procesador *quad-core* Intel Core i7-3610QE @2.3GHz, con controlador embebido PXI Express de un ancho de banda entre 4GB/s y 8 GB/s por ranura [8]. Además, el procesador se encuentra dispuesto en un chasis de 8 ranuras, que permite comunicarlo con módulos de adquisición de datos (DAQ), comunicación serial (hasta 16 puertos), entradas y salidas digitales, entre otros. Una ventaja de esta plataforma es que soporta programación paralela y en tiempo real por medio del módulo de programación de LabVIEW RT, así como la interacción con otros procesos y con el usuario, por medio del sistema operativo Windows 7.



**Figura 3.1:** Plataforma NI PXIe 8135 [8].

Para la implementación y configuración experimental, el módulo PXIe deberá comunicarse con la fuente de alta tensión CD, por medio de una de las entradas analógicas del su módulo de adquisición de señales analógicas NI PXIe 6363, a una frecuencia de muestreo de 2MS/s. Adicionalmente, tanto las lecturas de presión en la cámara, como la comunicación con la fuente de medición (*SMU*), para generación de las rampas según el método de Langmuir, será por medio de protocolo RS-232.

Como paso final, será importante extraer los parámetros de la red diseñada y verificada, para trasladar su funcionamiento a un módulo de instrumentación virtual (*VI*). Un aspecto fundamental en dicha implementación será la incorporación de las rutinas, y la red como tal, bajo la modalidad de tiempo real, utilizando las herramientas que provee LabVIEW para tal fin.





# Capítulo 4

## Identificación de la temperatura electrónica por medio de redes neuronales artificiales

En este capítulo se describe detalladamente la solución desarrollada para la identificación de la temperatura electrónica para plasmas fríos, por medio de la técnica de redes neuronales artificiales. En la sección 4.1 se presenta el diseño relacionado con el proceso de extracción y procesamiento de los datos de tensión-corriente para el posterior entrenamiento. La sección 4.2 presenta la selección de las topologías de red a utilizar. La determinación de las características de las redes neuronales artificiales a utilizar se muestra en la sección 4.3. Finalmente, la implementación del prototipo de las redes se presenta en la sección 4.4.

### 4.1 Extracción de datos

Como ya se describió previamente, la extracción de datos consiste en el desarrollo de experimentos en los que se genera un plasma CD en la cámara de vacío y por medio de la sonda de Langmuir dentro de él, se genera una rampa de tensión desde un potencial negativo (típicamente -50V) hasta un potencial positivo (típicamente 50V), y se mide la corriente generada en el circuito conformado por la fuente SMU, la sonda y el plasma. Los datos extraídos corresponden a una relación punto a punto de los pares de tensión CD de polarización y corriente generada en la sonda. Para el proceso de la extracción de los datos, como se mencionó previamente, se realizó una serie de modificaciones a la aplicación desarrollada por [25]. Esta aplicación se desarrolló en el entorno LabVIEW, originalmente para ser llevado a cabo por una plataforma NI CompactRIO. Una de las modificaciones realizadas a la aplicación, consistió en llevar el módulo de instrumentación virtual (*VI*) de forma que fuera compatible con la nueva plataforma NI PXIe. Otro de los aspectos modificados, corresponde a la sincronización en la toma de datos y la cantidad de

puntos de tensión-corriente adquiridos por el instrumento. En la versión anterior existía un límite fijo de caracteres en la recepción del puerto serial. La naturaleza estática de esta versión impedía obtener mediciones relevantes, ya que los rangos de medición también fueron estáticos. Otro de los cambios fue, por lo tanto, implementar la recepción de datos dinámicamente, conforme a los rangos de inicio y fin de tensión de polarización en la sonda. Adicionalmente, la versión anterior generaba un retardo innecesario entre puntos, lo que producía retrasos en la toma de los datos. Con esta corrección, la rampa de tensión puede realizarse en menos de 3 minutos, para un rango completo de -50V a 50V, proceso que anteriormente tomaba más de 5 minutos.

En el proceso de extracción de datos, se obtuvieron las curvas corriente-tensión, propias del método de Langmuir, para 65 experimentos distintos. En dichos experimentos se modificaron los parámetros de entrada; la presión varió entre los 30.0mTorr y los 90.0mTorr, la tensión CD aplicada se estableció para cuatro casos distintos: 0.5, 1, 1.5 y 2 KV, y para la posición de la sonda, se utilizaron tres posiciones distintas: 165, 170 y 175mm. En todo momento se utilizó Argón (Ar) como tipo de gas para los experimentos. La presión base de inicio de experimento fue de 0.1mTorr.

### 4.1.1 Cálculo de la temperatura electrónica

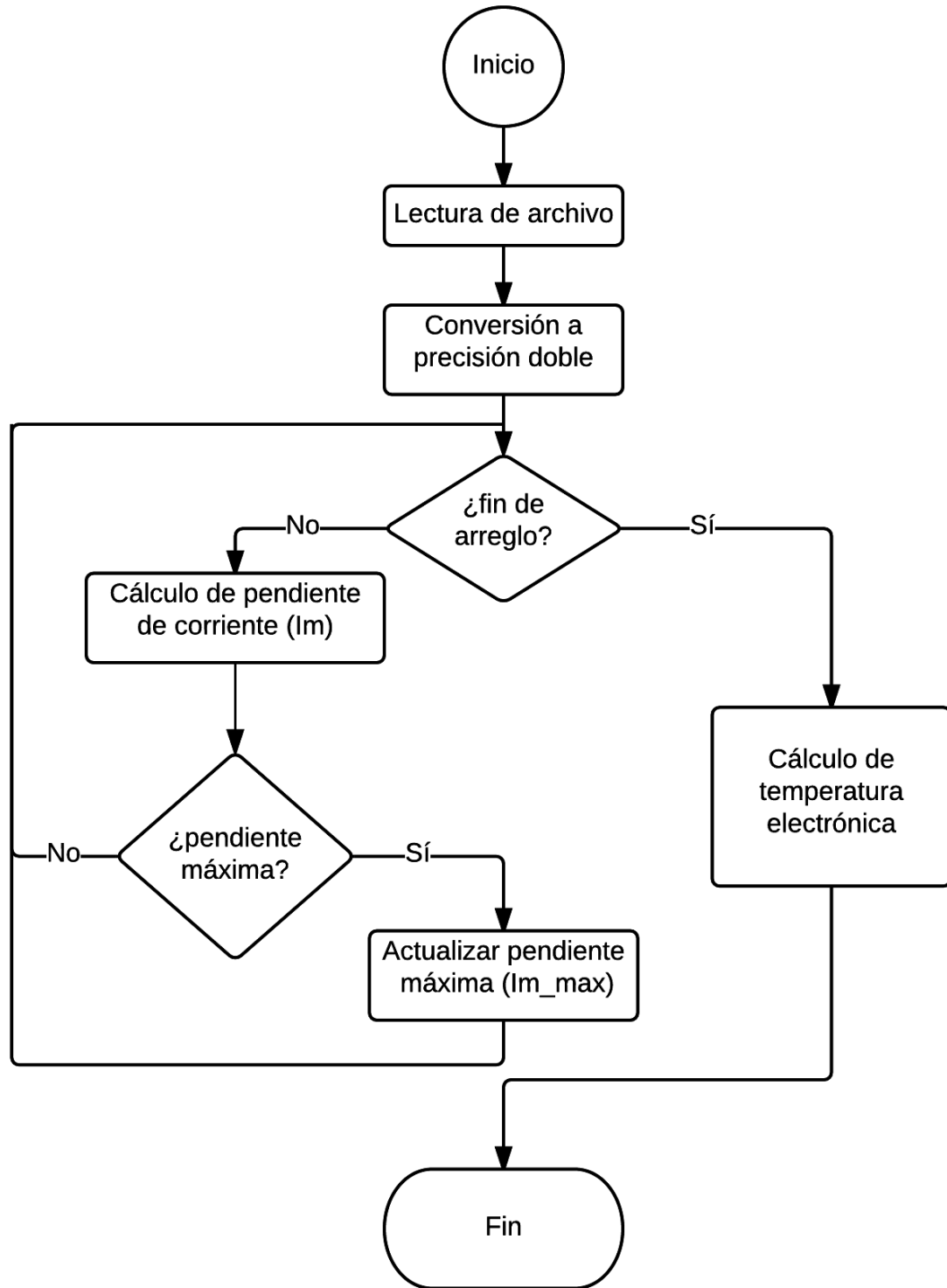
Una vez extraídos los pares corriente-tensión, fue necesario aplicar un análisis sobre las señales, de forma que se pudiera calcular la temperatura electrónica que identifica cada experimento, eso con el fin de generar pares entrada-salida para el posterior entrenamiento de la red. Con el fin de realizar este cálculo de la manera más automatizada posible, se diseñó una aplicación en C que permite la lectura de los archivos generados por la aplicación en LabVIEW y realiza el cálculo de la temperatura electrónica. En la Fig.4.1 se presenta el diagrama de flujo de la aplicación. Como se puede apreciar, el proceso inicia con la lectura del archivo, en el que se debe convertir cada valor de tensión y corriente a un valor numérico de doble precisión (64 bits), esto con el fin de no perder información significativa por cuantización, considerando que cada bit que se añade en la resolución aumenta en 6dB aproximadamente la potencia de la señal.

Una vez convertidos los puntos de tensión y corriente, y almacenados en un arreglo, se procede a calcular punto a punto la derivada discreta de la corriente, por medio del método de cálculo de la pendiente entre dos puntos consecutivos como:

$$\frac{dI_e}{dV_B} = \frac{I_e[n] - I_e[n-1]}{V_B[n] - V_B[n-1]} \quad (4.1)$$

Este cálculo es necesario ya que según la ecuación 2.1, la relación entre la corriente y su derivada con respecto a la tensión de polarización, en el punto en que su derivada es máxima (potencial del plasma), es igual a la temperatura electrónica, en eV.

Esto implica que además del cálculo de la derivada, se debe obtener el punto (índice en caso de los arreglos) en el que esta es máxima, este punto determina entonces el potencial



**Figura 4.1:** Diagrama de flujo del cálculo de la temperatura electrónica según método de Langmuir.

del plasma. En el diagrama de flujo se puede apreciar que luego de actualizar la pendiente máxima en cada iteración, al finalizar, se tendrá la derivada máxima de la corriente y el punto (índice) en el que esta ocurre. De esta forma, y como paso final, se procede a calcular la temperatura electrónica por medio de la implementación discreta de la ecuación 2.1. Dado que la pendiente máxima es compartida para los dos puntos consecutivos, para la determinación de la temperatura, se utiliza el valor medio de los puntos en los que la pendiente es máxima, como sigue:

$$eT = \frac{0.5 * (I[i_{max}] + I[i_{max} - 1])}{m_{max}} \quad (4.2)$$

donde  $i_{max}$  corresponde al índice superior del arreglo en el que la pendiente de la corriente fue máxima y  $m_{max}$  corresponde al máximo de esa pendiente.

## 4.2 Selección de topologías de red

Como se mencionó en la sección 2, existen diferentes clases de topologías de redes neuronales artificiales. Se debe tomar en cuenta para la decisión de la topología que, desde el punto de vista de aplicación, se requiere que la complejidad computacional no sea muy elevada, de forma que se puedan obtener los datos de temperatura electrónica dentro de un tiempo límite (*deadline*) de tiempo real. En este aspecto, es importante destacar las características de la plataforma de hardware donde se implementará el prototipo final. Como se discutió previamente, un aspecto clave es que la arquitectura del hardware es multiprocesador (hasta 4 núcleos). Otro aspecto relevante es la capacidad que tiene la plataforma de software LabVIEW para distribuir las tareas a diferentes procesadores, así como establecer requisitos de tiempo real. Con esta información, se procede a analizar los diferentes tipos de red descritos en la sección 2.

Si bien, las redes retroalimentadas o recurrentes (*RNN*), como el caso de las tipo Hopfield, ofrecen una ventaja significativa con respecto a las demás topologías, en términos de aprendizaje (no requieren un aprendizaje supervisado), como se puede apreciar en la Fig.2.6 existe una dependencia interna en la red entre las salidas anteriores y sus entradas. A pesar del dinamismo en el entrenamiento, esta dependencia desfavorece el nivel de paralelismo que se puede alcanzar. Disminuir el paralelismo afecta negativamente el desempeño en tiempo de ejecución (cumplimiento de *deadlines*), además de que no se aprovecharía los recursos de hardware, ya que al existir dependencias, la distribución de tareas (neuronas) por procesador, en escenarios de memoria compartida, puede provocar condiciones de carrera, sino se maneja adecuadamente. Una posible solución es agregar mecanismos de sincronización, como los semáforos, pero esta solución, a su vez, afecta negativamente el tiempo de ejecución, por lo que no es aceptable para esta aplicación.

Con respecto a las redes *PCANN*, en este caso, dado que las entradas son señales unidimensionales, el preprocesamiento por análisis de componentes principales no vendría

a beneficiar en ninguna medida al entrenamiento o ejecución de la red. La red doble (*DANN*), presentada en [7], requiere una gran cantidad de neuronas, como se muestra en la Fig.2.7, ya que consiste en redes de perceptrón multicapa (*MLP*) interconectadas. Desde el punto de vista de eficiencia en tiempos de computación, este enfoque requiere entonces más cálculos, así como un mayor esfuerzo en entrenamiento. Otro aspecto importante es que el consumo de memoria es también mayor, ya que requiere el doble de variables para los pesos de las diferentes capas. Por estos motivos, tampoco se utilizará este tipo de red en esta investigación.

Según las referencias consultadas, las topologías de mayor uso, y con un mejor desempeño, son las de perceptrón multicapa (*MLP*) y las de función de base radial (*RBF*) por lo que estas serán las dos topologías a utilizar, con el fin de determinar cuál brinda mejores resultados para la aplicación en cuestión.

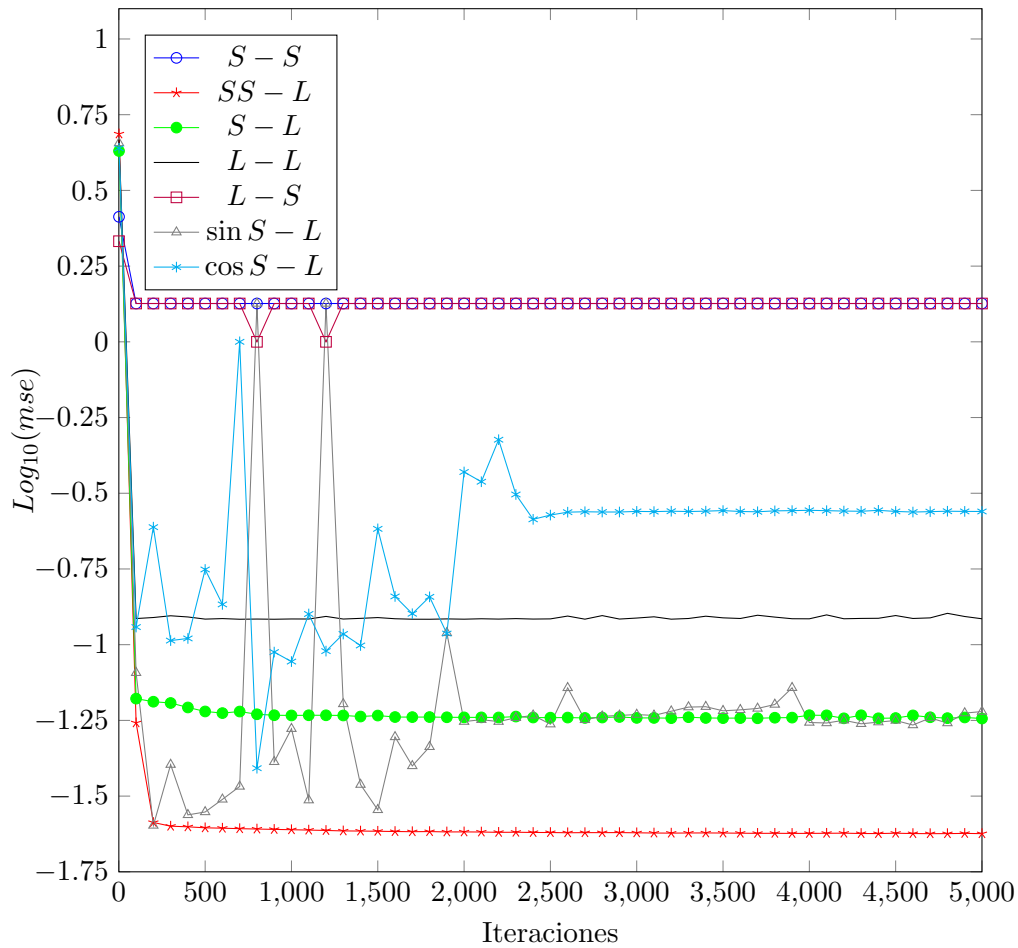
## 4.3 Definición de características de red neuronal

Como de definió previamente, se realizará una comparación entre dos tipos de red neuronal artificial, una del tipo perceptrón multicapa y otra del tipo de función de base radial. A continuación se describe el proceso para la selección de las características de cada una de las topologías, esto es: tipos de función de activación, número de capas y número de neuronas.

### 4.3.1 Perceptrón Multicapa

Uno de los puntos principales en la topología de red es la definición de las funciones de activación a utilizar. Previamente se describieron algunas de las funciones de activación más utilizadas en el ámbito de la identificación paramétrica con redes neuronales artificiales. Para el caso de la red de perceptrón multicapa (*MLP*), se debe determinar la función de activación de la(s) capa(s) oculta(s) y de la capa de salida. Para esto, se desarrolló un experimento en el que, basado en el entrenamiento de la red, se determinó el tipo de función de activación que más se adecua al problema de la identificación paramétrica de la temperatura electrónica del plasma.

El experimento consistió en entrenar la red *MLP* con 12 datos de entrenamiento de la forma entrada-salida, tomando como entradas la presión, tensión CD y posición de la sonda y, como salida, la temperatura electrónica, para diferentes funciones de activación, con un número de 10 neuronas y una única capa oculta. El algoritmo de entrenamiento utilizado fue el de *retropropagación* con 5000 iteraciones. La red y el entrenamiento se implementan por medio de la biblioteca *FANN*. En la Fig.4.2, se muestran los resultados del proceso de entrenamiento. Para facilitar la interpretación, se muestra el logaritmo del error cuadrático medio, en función del número de iteraciones. Para este experimento, se utilizaron 7 configuraciones de funciones de activación de capa oculta y capa de salida.



**Figura 4.2:** Logaritmo del error cuadrático medio contra número de iteraciones de entrenamiento para diferentes funciones de activación en red *MLP*.

Las configuraciones son (en forma de capa oculta-capas salida): sigmoideal-sigmoideal (S-S), sigmoideal simétrica-lineal (SS-L), sigmoideal-lineal (S-L), lineal-lineal (L-L), lineal-sigmoideal (L-S), senoideal simétrica-lineal (sinS-L), cosenoideal simétrica-lineal (cosS-L).

Como se puede apreciar, el menor error, con la convergencia más rápida, la presenta la configuración dada por la función de activación sigmoideal simétrica (forma de tanh) para la capa oculta y función de activación lineal para la capa de salida. Este resultado está de acuerdo a lo presentado en la sección 2. Además, dada la naturaleza no-lineal del plasma, se puede extraer de estos resultados, que la función de activación de capa oculta debe ser no-lineal, mientras que la función de activación de capa de salida debe ser lineal, para tener la capacidad de mapear en todo el rango de valores y no únicamente entre 0 y 1. Para el caso de esta configuración, el error de entrenamiento fue del 0.023.

## Determinación de la cantidad de capas y neuronas ocultas

La red neuronal *MLP*, como su nombre lo indica, posee múltiples capas de neuronas. Desde el punto de vista conceptual, este tipo de red requiere al menos una capa oculta, pero, dependiendo de la aplicación, puede requerir dos o incluso tres de ellas. Se debe considerar que aumentar el número de capas ocultas inherentemente aumenta el número de neuronas de la red, así como la carga computacional (implica más cálculos), pero puede dar mejores resultados, dependiendo de la aplicación.

Para la determinación del número de neuronas en la capa oculta de la red *MLP* se realizó un experimento en el que se varió la cantidad de neuronas de la capa y se procedió a entrenar con el set completo de datos (59), a una cantidad fija de 10000 iteraciones. Sobre cada experimento se obtiene el error cuadrático medio (*mse*); para determinar cuál configuración es óptima para la aplicación, se ideó una función de costo, que relaciona la cantidad de neuronas totales y el error:

$$FC = 0.7 * \frac{mse}{MAX(mse)} + 0.3 * \frac{N}{MAX(N)} \quad (4.3)$$

en donde *mse* corresponde al error cuadrático medio de cada configuración y N la cantidad de neuronas totales de la misma. En esta función, se asigna un peso de 70% al error en entrenamiento y un 30% a la cantidad de neuronas. De esta forma, la configuración que minimice la función de costo, será la óptima para la aplicación, es decir, posee el menor error, utilizando la menor cantidad de recursos (neuronas), lo que implica un menor tiempo de procesamiento. Las diferentes configuraciones de capas, neuronas, *mse* y función de costo se muestran en el anexo A.1. En la Fig.4.3 se muestra gráficamente el contenido de la tabla. Como se puede apreciar, la configuración que minimiza la función de costo corresponde a la número 4, que corresponde a 1 capa oculta con 4 neuronas en la misma. De esta manera, dicha configuración será la utilizada para el entrenamiento final y la validación cruzada de la red.

### 4.3.2 Función de base radial

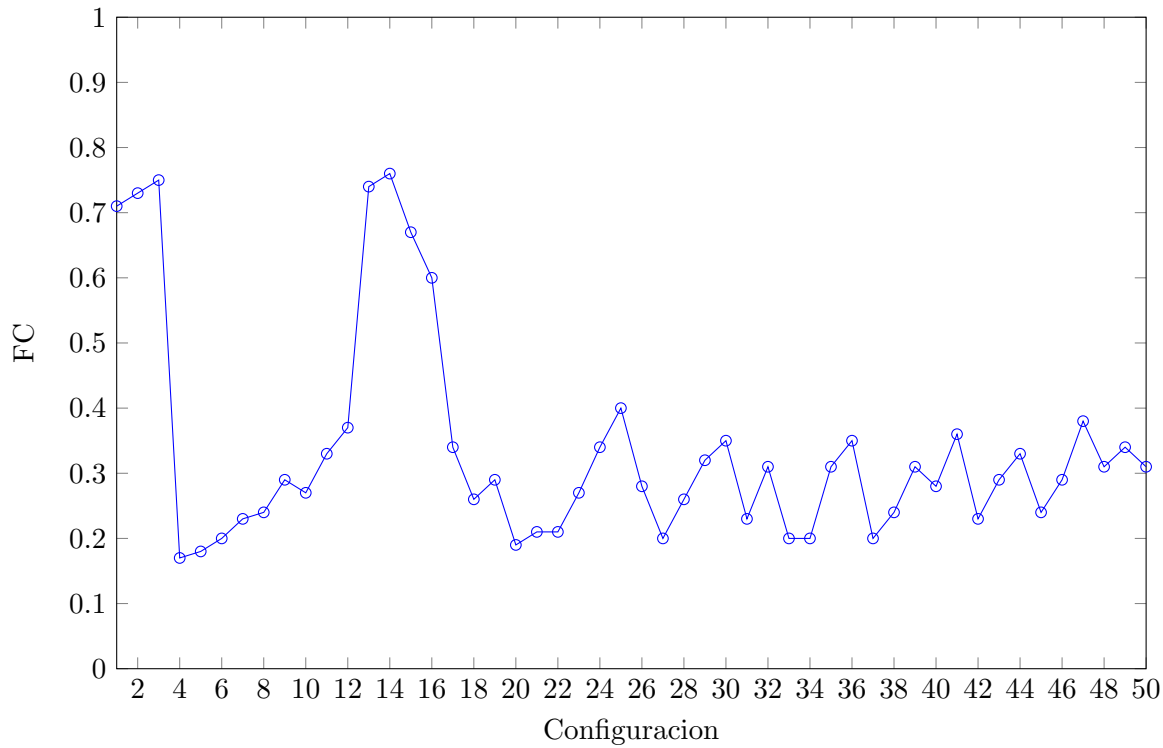
De forma similar a la red de perceptrón multicapa, se diseñó un experimento para determinar la función de activación radial más apropiada para la aplicación. La biblioteca *FANN* posee dos funciones de activación radiales: gaussiana y gaussiana simétrica. La función de activación gaussiana tiene la forma

$$y = e^{(-\alpha r)^2} \quad (4.4)$$

La función de activación gaussiana simétrica, por su parte, viene dada por:

$$y = 2e^{(\alpha r)^2} - 1 \quad (4.5)$$

La red de función de base radial (*RBF*), como se discutió previamente, posee una capa oculta, de función de activación radial y una capa de salida de función de activación



**Figura 4.3:** Función de costo para diferentes configuraciones de red *MLP*.

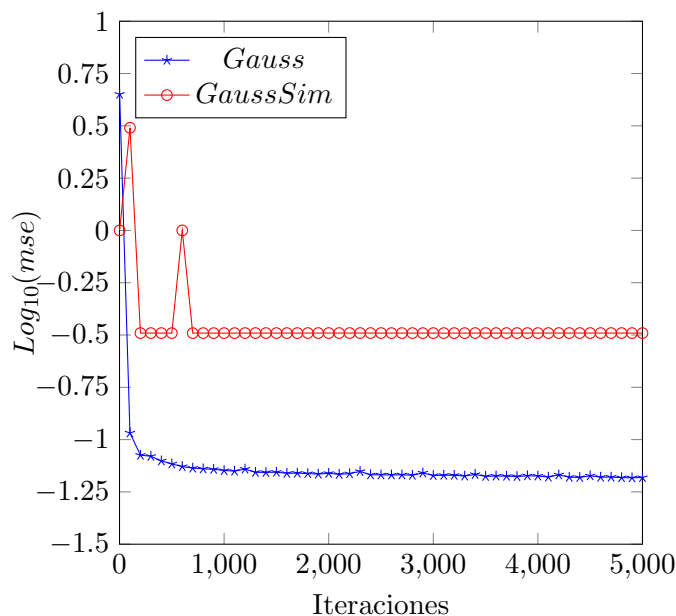
lineal, típicamente. En el experimento diseñado para determinar la función de activación más adecuada, de manera similar al *MLP*, se entrenó la red hasta un número de 5000 iteraciones para las dos funciones descritas, con 12 pares de entrada-salida. En la Fig.4.4 se muestran los resultados de dicho entrenamiento. Como se puede apreciar, la función de activación gaussiana posee un error menor a la gaussiana simétrica, además de converger más rápido, por lo que es más adecuada para la aplicación. En el caso de la función de activación gaussiana, el error cuadrático medio de entrenamiento fue de 0.065.

### Definición de número de neuronas

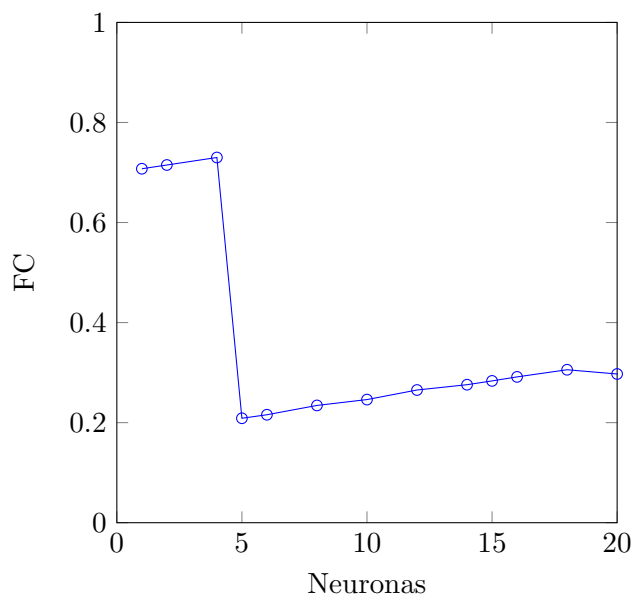
Como se mencionó previamente, la red de función de base radial, conceptualmente, posee una única capa oculta. En este tipo de red, es interesante la determinación del número de neuronas en esa capa, de forma que el error sea mínimo con la menor cantidad de neuronas, por ende procesamiento, posible.

De manera similar a la red *MLP*, se utilizó la función de costo de la ecuación 4.3. Para la red de función de base radial, se realizaron 15 configuraciones distintas de números neuronas en capa oculta. La tabla con el resultado de la función de costo para cada configuración se muestra en el anexo A.2. A modo de resumen, en la Fig.4.5, se muestran gráficamente los resultados de la función de costo. Para este caso, la configuración óptima corresponde a 5 neuronas en capa oculta.





**Figura 4.4:** Logaritmo del error cuadrático medio contra número de iteraciones de entrenamiento para diferentes funciones de activación en red *RBF*.



**Figura 4.5:** Función de costo para diferente número de neuronas de capa oculta en red *RBF*.

## 4.4 Implementación de prototipos

El paso final, en la validación de las redes neuronales artificiales como método de identificación paramétrica en el plasma, fue la implementación de prototipos que permitan un análisis de los tiempos de computación, y cumplimiento de *deadlines*, en diferentes plataformas: un computador de propósito general, un dispositivo PXIe 8135 y una plataforma

de evaluación de prototipos BeagleBone Black.

## Implementación en computador de propósito general

En el caso del computador de propósito general Dell Optiplex 7010, se realizó una implementación en lenguaje C de las redes MLP y RBF extraídas directamente del proceso de entrenamiento, por medio de la biblioteca *FANN*. En el anexo B se presentan los archivos de la red, utilizados como base para la implementación. Desde el punto de vista de la aplicación, se debe crear la estructura de tipo *fann* a partir del archivo generado en el entrenamiento:

```
struct fann *mlp = fann_create_from_file("mlp.net");
struct fann *rbf = fann_create_from_file("rbf.net");
```

para ambas redes. Posteriormente, se debe crear una variable de salida de tipo *fann\_type*, para almacenar el resultado de la ejecución de la red a partir de un arreglo con las entradas a ejecutar, como se muestra a continuación:

```
fann_type *mlpOut = fann_run(mlp, input);
fann_type *rbfOut = fann_run(rbf, input);
```

## Implementación en BeagleBone Black

En la plataforma BeagleBone Black (BBB), se siguió un proceso similar al anterior, sin embargo, ya que el sistema operativo fue construido a la medida, en la plataforma no se tienen las herramientas para la compilación nativa (compilador, ensamblador, enlazador, etc.), por lo que se debió seguir una metodología de compilación cruzada, por medio del *toolchain*, también construido con Yocto, y archivos *Makefiles* para tal fin, en una estación de desarrollo (*host*) diferente a la plataforma BBB. En este sentido cabe además destacar que se debió compilar (portear) la biblioteca *FANN* para la plataforma utilizando el *toolchain*. Otra diferencia fundamental entre las aplicaciones de la plataforma BBB y el computador de propósito general es que el sistema operativo, en el caso de la primera, fue construido con soporte para tiempo real. En la aplicación en tiempo real, se establece la prioridad de la tarea dentro del calendarizador (*scheduler*) del sistema, así como el algoritmo de calendarización para la misma, de la forma:

```
struct sched_param param;

/*Establece la prioridad máxima a la tarea con algoritmo FIFO*/
param.sched_priority = sched_get_priority_max(SCHED_FIFO);

/*Verifica que no haya errores en configuración de calendarizador*/
```

```
if(sched_setscheduler((pid_t) 0, SCHED_FIFO, &param) == -1) {  
printf("couldn't set scheduler to SCHED_FIFO\n");  
exit(-1);  
}
```

## Implementación en PXIe 8135

Para el caso de la plataforma PXIe 8135, se realizó una implementación propia de las redes neuronales artificiales, siguiendo los modelos presentados en la sección 2. Desde el punto de vista de plataforma, el lenguaje de programación LabVIEW aprovecha de manera natural las características de multiprocesador, mediante el paralelismo de tareas y de datos, para operaciones independientes. De esta forma, dos o más tareas independientes correrán en paralelo sin necesidad de técnicas de programación extra. En el caso de sistemas multiprocesador, cada tarea se ejecuta como un hilo independiente en un procesador diferente, incrementando el desempeño en tiempos de ejecución [27].

En las figuras 4.6 y 4.7 se muestran las implementaciones de las redes neuronales realizadas. Como se puede apreciar, cada neurona, encerrada en un cuadro, es totalmente independiente de las demás. Para el caso de la neurona de salida, comentada con la palabra *lineal*, sí hay dependencia de los resultados de las neuronas anteriores, por lo que no se ejecutará hasta que las tareas de las demás neuronas hayan finalizado. Dicha situación representa un mecanismo de sincronización de barrera y es deseable, de forma que la red funcione adecuadamente. Dentro de cada neurona se utiliza un *SubVI* que contiene la función de activación. Para el caso de la red MLP, la función de activación de las neuronas de capa oculta es *tanh*, mientras que para la red RBF se trata de función gaussiana, expresada en la ecuación 4.4.

Cabe destacar que los pesos en las neuronas, asignados mediante constantes, son aleatorios y no presentan necesariamente los determinados por el entrenamiento de la red por medio de la biblioteca *FANN*, esto ya que el objetivo de la implementación es la medición de tiempos ejecución de una red neuronal RBF y MLP genérica, tomando en cuenta la arquitectura dimensionada previamente.

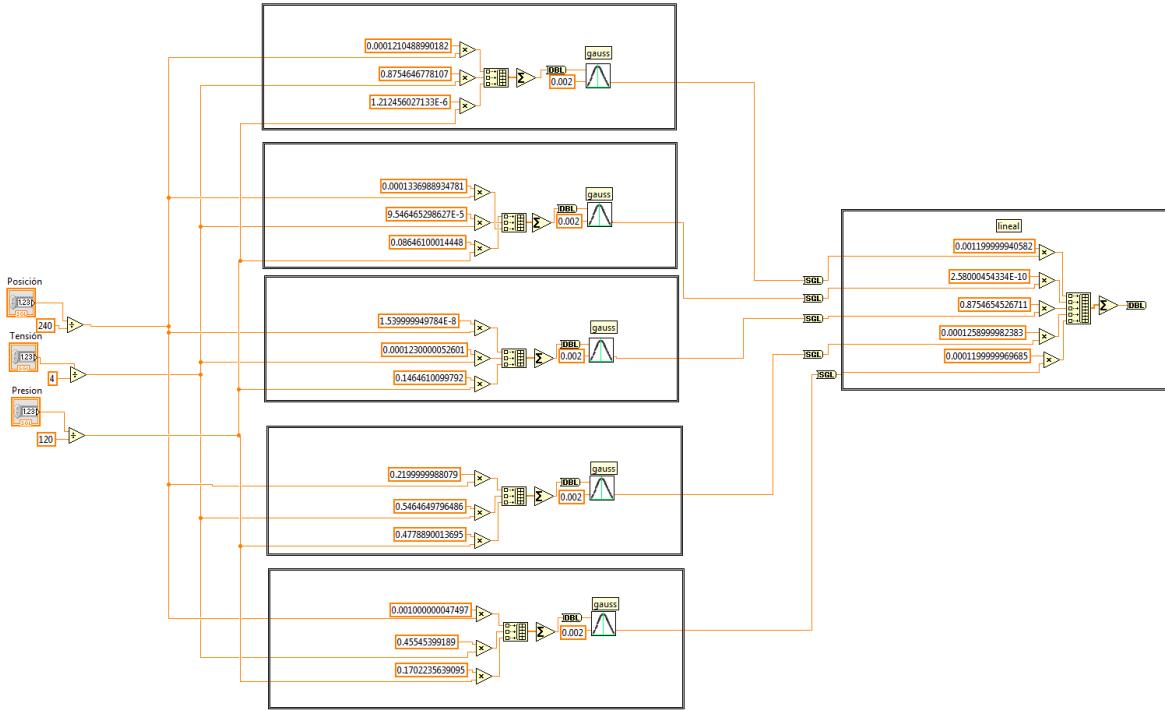


Figura 4.6: Red neuronal RBF en LabVIEW

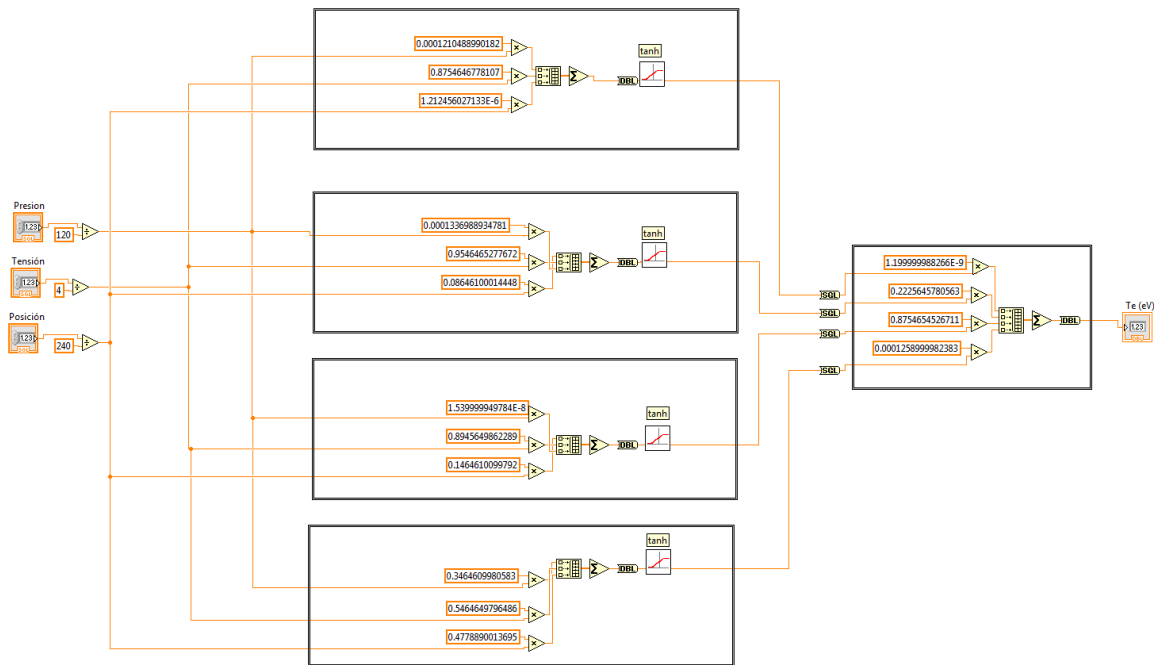


Figura 4.7: Red neuronal MLP en LabVIEW

# Capítulo 5

## Resultados y análisis

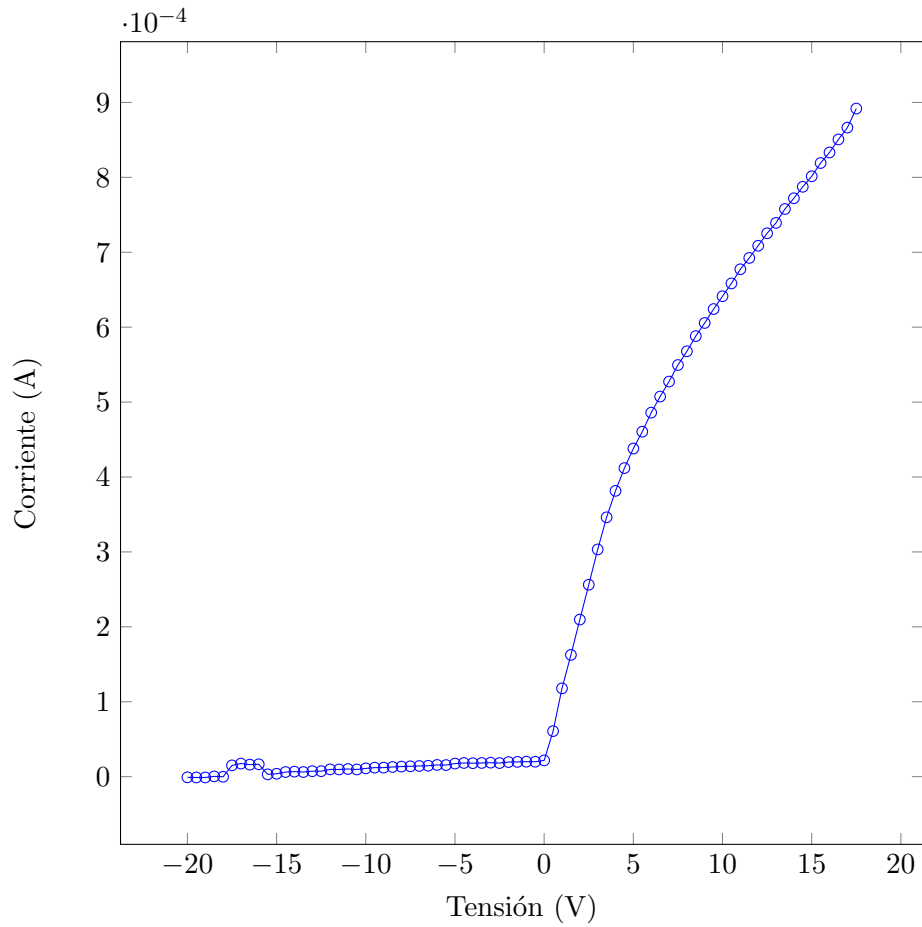
En esta sección se incluyen los principales resultados obtenidos a partir del trabajo desarrollado, así como un análisis de los mismos. Se abarcan temas de extracción experimental de datos, cálculo algorítmico de temperatura electrónica, entrenamiento y validación de las redes neuronales y los detalles de implementación en sistemas de tiempo real.

### Extracción experimental de datos

El punto inicial de este trabajo se basa en la extracción experimental de las curvas tensión-corriente, provenientes del diagnóstico de Sonda de Langmuir. Para tal fin, se trabajó con base al desarrollo parcial de la aplicación previamente diseñada por [25]. En la figura 5.1 se muestra una curva típica del diagnóstico de Langmuir, extraída como parte del set completo de datos. Como se ha descrito anteriormente, estos datos servirán como entrada para el cálculo algoritmo de la temperatura electrónica y el posterior entrenamiento y validación de las redes neuronales.

### Cálculo algorítmico de la temperatura electrónica

Como parte de los primeros resultados, se realizó una prueba del algoritmo de cálculo de la temperatura electrónica, ya que este preprocesamiento es clave para el entrenamiento de la red neuronal. Si este cálculo es incorrecto, la red estaría entrenada incorrectamente, lo que produciría errores en su funcionamiento. Para verificar el funcionamiento del algoritmo diseñado, se utilizó un set de datos de prueba externo, disponible en [9]. En dicho set, se obtienen 2500 puntos de pares corriente-tensión en la sonda. La ventaja de utilizarlo como mecanismo de verificación es que posee el valor de temperatura electrónica calculado (tomado como valor teórico) previamente. A pesar de que los métodos de cálculo para la temperatura electrónica entre el algoritmo diseñado y la implementación en el set externo son distintos, como se muestra en la Tabla 5.1, el porcentaje de error entre ambas es menor al 1%, lo que permite la validación del algoritmo. En este punto cabe



**Figura 5.1:** Curva típica de corriente-tensión obtenida experimentalmente en sonda de Langmuir.

además destacar que la implementación del algoritmo se realizó de forma que fuera lo más autónoma posible, mientras que la implementación realizada en [9] es manual y empírica, lo que no permite afinar sustancialmente en los cálculos.

**Tabla 5.1:** Comparación entre  $T_e$  obtenida algorítmicamente y referencia externa, con datos de prueba de [9].

	Valor obtenido	Valor de referencia	Error (%)
Te (eV)	3.60	3.57	0.8

Como segunda opción de validación, por medio de los recursos disponibles en el laboratorio virtual *e-lab* [28], se llevó a cabo un experimento que replicara las condiciones presentadas por [29]. En dicho experimento, se realiza una descarga de plasma de una mezcla de gases

de He y Ne, por radio frecuencia (RF), a una presión base de 0.45mBar. Sobre los datos obtenidos en el experimento, se aplicó el algoritmo para la obtención de la temperatura electrónica. En la tabla 5.2 se muestra el resumen y comparación entre el resultado obtenido por el algoritmo y el resultado presentado por [29], utilizado como referencia. Como se puede apreciar, el error entre ambos es menor al 5%.

**Tabla 5.2:** Comparación entre  $T_e$  obtenida algorítmicamente y referencia externa, con datos obtenidos en experimento en *e-lab* .

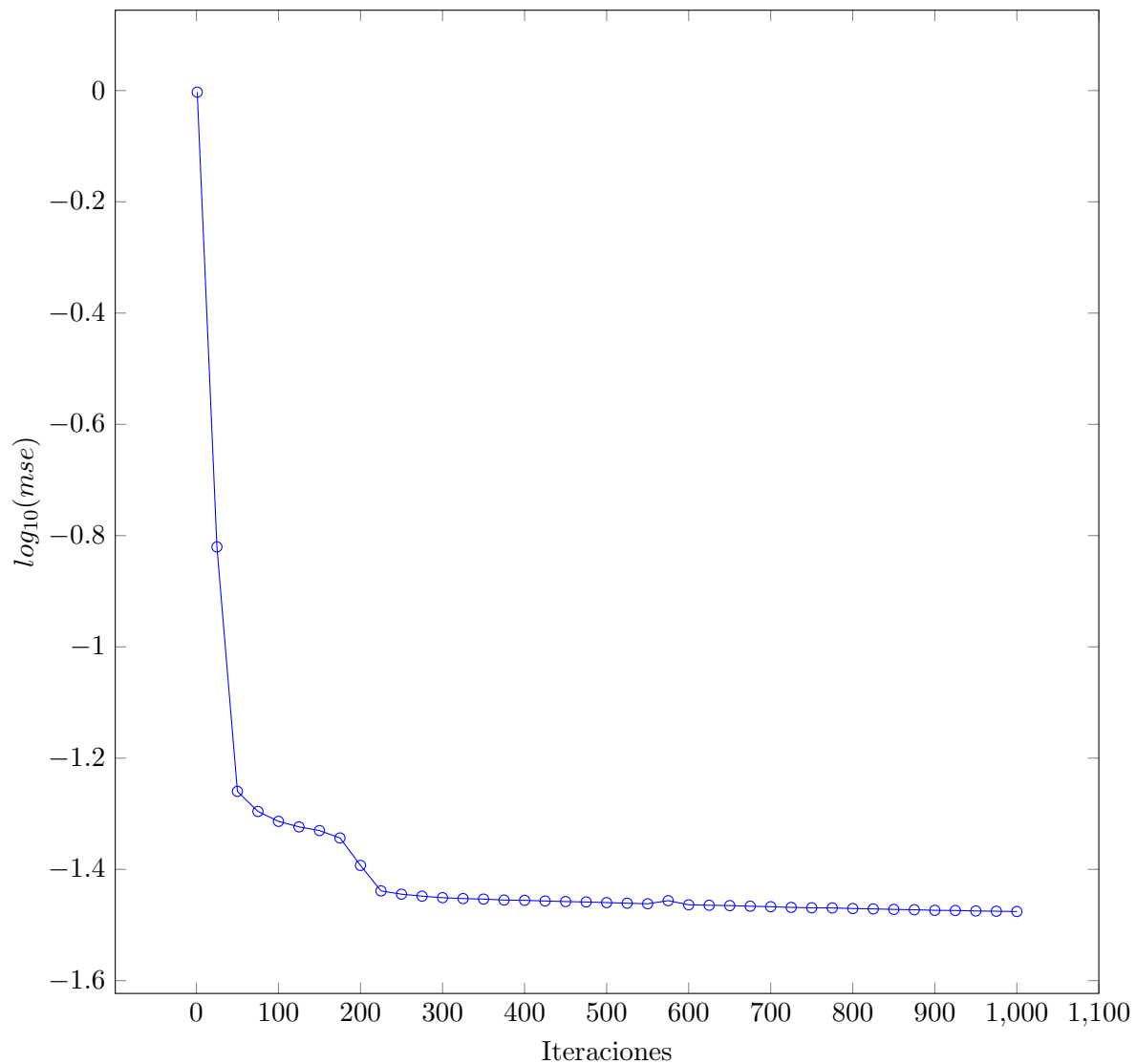
	Valor obtenido	Valor de referencia	Error (%)
Te (eV)	11.91	12.37	3.7

Como tercera opción de validación del método, por medio cualitativo, se toma la opinión de experto del Dr. Horacio Fernades, del Instituto Técnico de Lisboa, quién coordina el laboratorio virtual *e-lab*, que determinó que el método aplicado es “robusto” y “ampliamente utilizado” para el cálculo de la temperatura electrónica. Además, con respecto a la réplica del experimento realizado, determinó que el valor se encuentra dentro lo esperado para descargas del tipo RF.

## Entrenamiento de las redes neuronales

El siguiente paso corresponde al entrenamiento de las redes neuronales artificiales. Dicho entrenamiento se realiza con 59 datos, de la forma entrada-salida. En la Fig.5.2, se muestra la curva de entrenamiento para las primeras 1000 iteraciones para el caso de la red MLP. Como se puede observar, a partir de la iteración 300 aproximadamente, el logaritmo (mostrado para facilitar de apreciación) del error decrece lentamente. En 1000 iteraciones, el logaritmo del error cuadrático medio ha descendido a  $-1.47572$ , lo que implica un *mse* de 0.03344, aproximadamente. Para un total de 33501 iteraciones, el error cuadrático medio es 0.03. Porcentualmente, este error se encuentra en el rango establecido como aceptable del 10%. Dado el comportamiento estable para la mayor parte de las iteraciones, se puede extraer que con base en esta configuración de red, aumentar el tiempo de entrenamiento no mejora sustancialmente desempeño de la red.

Con respecto a la red *RBF*, cuya curva de entrenamiento se muestra en la Fig.5.3, de manera similar a la red *MLP*, se realizó un entrenamiento con 59 pares de datos entrada-salida. Como se puede apreciar, la curva del entrenamiento converge más lento que la de la red de perceptón multicapa. Asimismo, el error cuadrático medio es mayor. De hecho, similar al caso anterior, a partir de las 400 iteraciones el logaritmo del error se estabiliza en un valor aproximado de 1.335, equivalente a un *mse* de 0.0441. Para un total de 40000



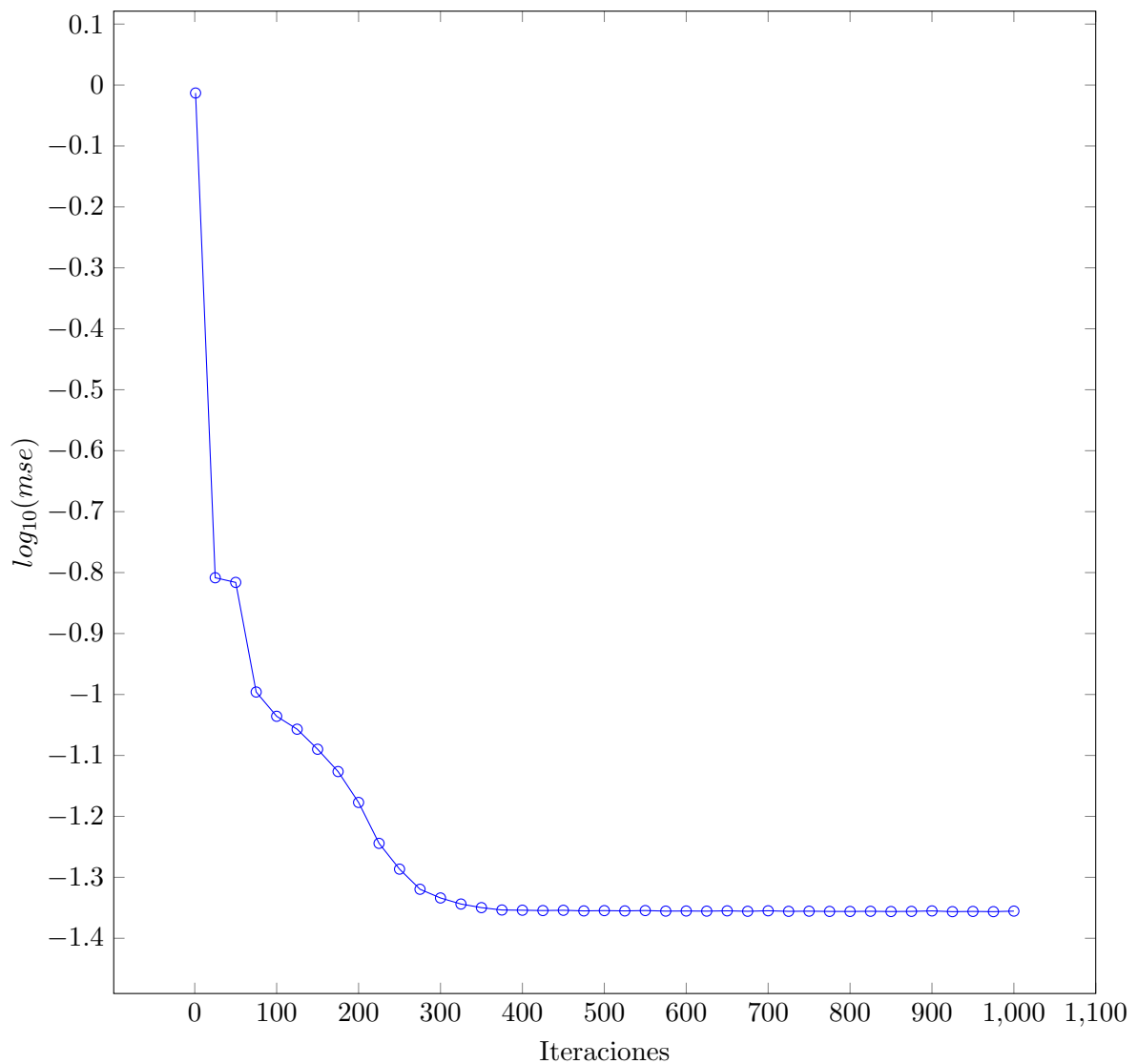
**Figura 5.2:** Curva de entrenamiento de red MLP.

iteraciones, el error cuadrático medio es 0.0437, lo que, además de ser mayor que la red MLP, implica además que aumentar las interacciones en el entrenamiento tampoco viene a mejorar significativamente el desempeño de la red. Desde el punto de vista porcentual, sin embargo, el error se encuentra en el margen considerado como aceptable.

## Validación de las redes neuronales

Posterior al entrenamiento, se realizó la validación cruzada de las redes neuronales artificiales dimensionadas. Para tal fin, se determinó un porcentaje del 10% del set de datos completo como datos de validación. Cabe destacar que los datos se seleccionaron aleatoriamente y se aplicaron sin ninguna modificación para las dos redes a validar. Las tablas 5.3 y 5.4 muestran los resultados de la validación cruzada para las redes perceptrón





**Figura 5.3:** Curva de entrenamiento de red RBF.

multicapa y función de base radial respectivamente. Para el caso de la red MLP, 4 de los 6 datos de validación poseen un porcentaje de error (absoluto) menor al 10%. Para el caso de la RBF, solamente uno de los 6 datos de validación se sale del rango del 10% de error. Desde el punto de vista promedio, como se muestra en la tabla 5.5, ambas redes tienen un porcentaje de error de validación promedio menor al 10%, valor considerado como aceptable para la etapa de validación.

Dichos resultados demuestran que el método de redes neuronales, para el caso de las redes MLP y RBF, es válido para la identificación de la temperatura electrónica del plasma. Cabe destacar que el método podría mejorarse si los tiempos de computación no fueran una limitante, es decir, si la función de costo de la ecuación 4.3 no fuera aplicable, el error podría disminuir, al aumentar el número de neuronas en las redes. Sin embargo, dada la naturaleza de la aplicación, en la que se debe obtener la temperatura en tiempo real, para descargas de plasma cuya duración se encuentra en el rango de las decenas

de milisegundos, la función de costo aplicada permitió que las redes tuvieran un error promedio aceptable, optimizando los recursos y tiempos de computación.

**Tabla 5.3:** Validación cruzada de red MLP

Entradas			Salida esperada	Salida MLP	Error
Presión (mTorr)	Tensión (KV)	Posición (mm)	Te (eV)	Te (eV)	% error
71.2	1.5	160	1.425887	1.497575	5.04
71.8	2	165	1.519697	1.361137	10.4
60.9	1	160	1.166231	1.128585	3.22
83.8	1	170	1.429131	1.400900	1.97
70.8	0.5	165	0.599119	0.566427	5.45
51.5	2	160	1.534854	1.346853	12.2

**Tabla 5.4:** Validación cruzada de red RBF

Entradas			Salida esperada	Salida RBF	Error
Presión (mTorr)	Tensión (KV)	Posición (mm)	Te (eV)	Te (eV)	% error
71.2	1.5	160	1.425887	1.420741	0.36
71.8	2	165	1.519697	1.462064	3.79
60.9	1	160	1.166231	0.959851	17.7
83.8	1	170	1.429131	1.302636	8.85
70.8	0.5	165	0.599119	0.615093	2.66
51.5	2	160	1.534854	1.346853	6.76

**Tabla 5.5:** Error cuadrático medio y porcentaje de error absoluto en validación cruzada de las redes MLP y RBF.

Red	mse	% error promedio
MLP	0.011485	8.613643
RBF	0.012164	8.591420

Como se puede apreciar en las tablas anteriores, virtualmente, ambas redes tienen un desempeño equivalente. Un aspecto importante a destacar es que la red RBF posee una neurona más en capa oculta que la red MLP. De hecho, este factor contribuye a que el error promedio de la red RBF sea relativamente menor a la MLP. Desde el punto de vista del error cuadrático medio, la RBF posee un mayor error, esto ya que la desviación del único resultado fuera del rango del 10% es mucho mayor que cualquier desviación en la red MLP.

## Evaluación de plataformas en implementación

Como se describió previamente en la sección 3, como paso final de la aplicación de las redes neuronales artificiales a la identificación paramétrica en el plasma, se realizaron tres implementaciones en diferentes plataformas, para así evaluar el desempeño de las redes dimensionadas, en términos de tiempo de ejecución y cumplimiento de *deadlines* de tiempo real, en los casos con soporte para tal fin.

En la tabla 5.6 se muestran los resultados obtenidos de la ejecución de las dos redes en las diferentes plataformas. Para la determinación del tiempo de ejecución se realizaron 1000000 ejecuciones y se calculó el tiempo promedio de ejecución para las redes MLP y RBF. Como se puede apreciar, en las implementaciones del computador general y de la plataforma BeagleBone Black, por medio de la biblioteca *FANN*, la red RBF mantiene un mejor tiempo de ejecución que en la MLP. Por otro lado, en los prototipos de redes implementados manualmente en la plataforma NI PXIe, la red MLP posee menor tiempo de ejecución. Un aspecto fundamental es que para todas las plataformas, el tiempo se mantiene en el rango de los cientos y miles de nanosegundos, esto es relevante, ya que en el control en tiempo real de los parámetros del plasma, por ejemplo en una descarga típica para el dispositivo de confinamiento magnético del plasma MEDUSA-CR, se debe obtener el parámetro en tiempos menores a 15ms, ya que este es el tiempo de duración esperado. En este sentido, en una descarga de 15ms, se podrán ejecutar al menos 3200 cálculos de la temperatura electrónica por el método de las redes neuronales artificiales, considerando el peor de los casos de implementación en las diferentes plataformas. Cabe destacar que el método anterior requería la generación de las curvas y cálculo de la temperatura fuera de línea, lo que hacía imposible algún control en tiempo real del plasma.

**Tabla 5.6:** Tiempo de ejecución de las redes neuronales artificiales en diferentes plataformas

Plataforma	# proc.	Frec. operación (GHZ)	Tiempo real	Tiempo ejecución (ns)	
				MLP	RBF
Dell Optiplex 7010	4	3.1	No	260.9	228.6
NI PXIe 8135	4	2.3	Sí Labview RT	448.55	488.80
BeagleBone Black	1	1	Sí linux rt-kernel	4578	4565

La diversidad de las plataformas permite evaluar el método en tres escenarios distintos. En el primer caso, de la plataforma Dell Optiplex, se posee el menor tiempo de ejecución, pero dado que corresponde a un sistema de propósito general, no presenta utilidad en

la aplicación específica, por lo que se utiliza solamente como referencia. La plataforma, correspondiente al dispositivo NI PXIe, representa un escenario ideal, ya que posee soporte para el procesamiento paralelo en sus múltiples núcleos y tiempo real, además de poseer diferentes módulos para interacción con entradas y salidas. En el caso de la plataforma BeagleBone Black, se demuestra el funcionamiento de las redes y el cumplimiento de tiempos de ejecución aún en el escenario en que se poseen recursos más limitados. Esta implementación se planteó de forma que el sistema operativo solo tiene los elementos necesarios para la ejecución de la aplicación, por lo que es óptimo en consumo energético (no hay tareas innecesarias siendo ejecutadas por el CPU), pero su desventaja es que no posee el suficiente soporte para paralelismo real, ya que solo posee un procesador. Además, dado su bajo costo, sirve como un sistema de respaldo para la plataforma PXIe, así como una buena alternativa para futuras implementaciones.

En el caso de la plataforma PXIe y BeagleBone Black (BBB), que tienen soporte para tiempo real, se estableció un *deadline* (explícito para la primera, implícito para segunda) de 1ms, de forma que la tarea de ejecución de la red debe llevarse a cabo en menos de este tiempo. Para ambos casos, dados los tiempos de ejecución obtenidos, se da un cumplimiento al 100% de dicho *deadline* lo que valida esta aplicación de redes neuronales en tiempo real.

# Capítulo 6

## Conclusiones y recomendaciones

Los resultados obtenidos permiten determinar que el método de redes neuronales artificiales, como medio de extracción o identificación paramétrica, sí es aplicable para el caso de la temperatura electrónica del plasma. En este sentido, los tipos de redes más apropiadas para la aplicación fueron la red de percepción multicapa (MLP) y la red de función de base radial (RBF).

El proceso de entrenamiento, tanto para la red MLP como la RBF, se mantuvo en un rango de error esperado, lo que permitió no solamente el cumplimiento del objetivo planteado, sino también optimizar las características de las redes en base a este parámetro.

Desde el punto de vista de validación cruzada, ambas redes tuvieron un porcentaje de error promedio menor al 10%, lo que permite considerar el método como válido para la aplicación. A su vez, la red RBF tuvo menor error promedio y tuvo mayor cantidad de datos dentro del rango de error.

Considerando además los tiempos de ejecución, en el computador de propósito general y la plataforma BeagleBone Black, la red más apropiada para la aplicación de extracción es RBF. Por otra parte, en la plataforma NI PXIe, el prototipo red de MLP implementado posee menor tiempo de procesamiento que el RBF.

Desde el punto de vista de las plataformas, el dispositivo NI PXIe representó un escenario ideal para la implementación de las redes neuronales, de forma que se aprovechó la capacidad de paralelismo inherente de LabVIEW, mientras se mantuvo un tiempo de ejecución aceptable que cumple el requisito de plazo de tiempo (*deadline*) establecido de 1ms. En este escenario, cabe destacar que la plataforma de evaluación de prototipos BeagleBone Black, bajo un sistema operativo de tiempo real, es otra alternativa para la implementación del sistema, con la ventaja de que mantiene el desempeño en tiempo de ejecución a un bajo costo. Sin embargo, su capacidad de paralelismo es limitada.

## Recomendaciones

A fin de mejorar de mayor manera el tiempo de respuesta de las redes, se puede tomar en cuenta las técnicas de computación aproximada (*approximate computing*), específicamente para las funciones de activación. Desde el punto de vista de mejoras de tiempo de respuestas, en caso de que la complejidad de la red crezca, la función de costo planteada puede llegar a jugar un papel fundamental. En este sentido, puede requerirse una implementación puramente en hardware, en dispositivos FPGA. La función de costo planteada, en este escenario, es indispensable, al minimizar el espacio, y el tiempo de computación.

Con respecto a los datos que se salen del rango de error en la verificación cruzada, que pueden considerarse *outliers*, estos pueden reducirse al aumentar los datos el set de entrenamiento. De esta manera, un trabajo futuro requiere un mayor énfasis en el entrenamiento, aumentando el número de datos dentro del set total.

# Bibliografía

- [1] J. Gonzalez, C. Soto, J. Carvajal, and C. Ribeiro, “Energy, vacuum, gas fueling, and security systems for the spherical tokamak medusa-cr,” *Bulletin of the American Physical Society*, vol. 58, 2013.
- [2] I. Vargas, J. Mora, C. Otarola, J. Asenjo, E. Zamora, J. Gonzalez, and C. Piedra, “Constructing a small modular stellarator in latin america,” in *APS Meeting Abstracts*, vol. 1, p. 8034P, 2014.
- [3] R. L. Merlino, “Understanding langmuir probe current-voltage characteristics,” *American Journal of Physics*, vol. 75, no. 12, pp. 1078–1085, 2007.
- [4] T. J. M. Boyd and J. J. Sanderson, *The Physics of Plasmas*. Cambridge University Press, 2003.
- [5] C. M. Bishop, “Neural networks and their applications,” *Review of scientific instruments*, vol. 65, no. 6, pp. 1803–1832, 1994.
- [6] K. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” vol. 1, no. 1, pp. 4–27.
- [7] A. Sengupta and P. Ranjan, “Modified neural networks for rapid recovery of tokamak plasma parameters for real time control,” *Review of scientific instruments*, vol. 73, no. 7, pp. 2566–2577, 2002.
- [8] “PXIe-8135 - National Instruments.”  
<http://sine.ni.com/nips/cds/view/p/lang/en/nid/210545>.
- [9] D. Pace, “Example of Langmuir Probe Analysis.”  
<http://www.davidpace.com/physics/graduate-school/langmuir-analysis/5.htm>.
- [10] M. Ariola and A. Pironti, *Magnetic control of tokamak plasmas*. Springer Science & Business Media, 2008.
- [11] A. H. Boozer, “What is a stellarator?,” *Physics of Plasmas (1994-present)*, vol. 5, no. 5, pp. 1647–1655, 1998.

- [12] D. Hartmann, “Stellarators,” *Fusion Science and Technology*, vol. 49, no. 2T, pp. 43–55, 2006.
- [13] J. E. Harry, *Introduction to Plasma Technology: Science, Engineering, and Applications*. John Wiley & Sons, 2013.
- [14] A. Donné, A. Costley, R. Barnsley, H. Bindslev, R. Boivin, G. Conway, R. Fisher, R. Giannella, H. Hartfuss, M. Von Hellermann, *et al.*, “Diagnostics,” *Nuclear Fusion*, vol. 47, no. 6, p. S337, 2007.
- [15] C. Bishop, P. Haynes, M. Smith, T. Todd, and D. Trotman, “Real-time control of a tokamak plasma using neural networks,” *Neural Computation*, vol. 7, no. 1, pp. 206–217, 1995.
- [16] S. Nissen and E. Nemerson, “Fast artificial neural network library,” *Disponibile en leenissen.dk/fann/html/files/fann-h.html*, 2000.
- [17] H. M. Mott-Smith and I. Langmuir, “The theory of collectors in gaseous discharges,” *Physical review*, vol. 28, no. 4, p. 727, 1926.
- [18] M. Hopkins and W. Graham, “Langmuir probe technique for plasma parameter measurement in a medium density discharge,” *Review of scientific instruments*, vol. 57, no. 9, pp. 2210–2217, 1986.
- [19] R. Rajesh, B. R. Kumar, S. Varshney, M. Kumar, C. Chavda, A. Thakkar, N. Patel, A. Kumar, and A. Team, “Electron temperature ( $t_e$ ) measurements by thomson scattering system,” *Pramana*, vol. 55, no. 5-6, pp. 733–740, 2000.
- [20] N. Peacock, D. Robinson, M. Forrest, and P. Wilcock, “Measurement of the electron temperature by thomson scattering in tokamak t3,” *Nature*, vol. 224, pp. 488–490, 1969.
- [21] C. M. Bishop *et al.*, “Neural networks for pattern recognition,” 1995.
- [22] L. K. Hansen and P. Salamon, “Neural network ensembles,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 10, pp. 993–1001, 1990.
- [23] J. B. Lister and H. Schnurrenberger, “Fast non-linear extraction of plasma equilibrium parameters using a neural network mapping,” *Nuclear Fusion*, vol. 31, no. 7, p. 1291, 1991.
- [24] W. Teng, G. Xiang-Dong, and L. Wei, “Characterisation of the plasma density with two artificial neural network models,” *Chinese Physics B*, vol. 19, no. 7, p. 070505, 2010.
- [25] C. Soto, “Desarrollo de un sistema de adquisición y procesamiento de datos de los diagnosticos del plasma del scr-1: sonda langmuir e interferómetro heterodino.” tech. rep., Instituto Tecnológico de Costa Rica, 2013.



- 
- [26] s.a, “About yocto project.”  
<https://www.yoctoproject.org/about>.
- [27] N. Instruments, “Programming Strategies for Multicore Processing: Task Parallelism - National Instruments.”  
<http://www.ni.com/white-paper/6420/en/pdf>, Dec. 2011.
- [28] s.a, “What is e-lab?.”  
[http://www.elab.ist.utl.pt/?page\\_id=118](http://www.elab.ist.utl.pt/?page_id=118).
- [29] A. Freire, J. Lopes, and J. Loureiro, “Langmuir probe experiment in e-lab,” tech. rep., Instituto Superior Técnico, Universidade Técnica de Lisboa, s.f.



# Apéndice A

## Datos para determinación de número de capas y neuronas

A continuación se presentan los experimentos realizados para la determinación del número de neuronas y capas óptimas en las redes neuronales de perceptón multicapa y función de base radial.

### A.1 Perceptron multicapa

**Tabla A.1:** Función de costo para selección de número de neuronas de capa oculta en red neuronal MLP.

Configuración	Capas ocultas	Neuronas en C.O 1	Neuronas en C.O 2	mse	FC
1	1	1	0	0.1967714876	0.71
2	1	2	0	0.1967717111	0.73
3	1	3	0	0.1967719346	0.75
4	1	4	0	0.0308199301	0.16
5	1	5	0	0.0297542885	0.18
6	1	6	0	0.0308865439	0.20
7	1	8	0	0.03003929	0.23
8	1	10	0	0.0260783415	0.24
9	1	12	0	0.0297777131	0.29
10	1	14	0	0.0161112119	0.27
11	1	15	0	0.0296452101	0.33
12	1	20	0	0.0201802347	0.37
13	2	2	1	0.1967716664	0.74
14	2	2	2	0.1967714727	0.76
15	2	2	3	0.168155089	0.67

Continuación tabla A.1.

Configuración	Capas ocultas	Neuronas en C.O 1	Neuronas en C.O 2	mse	FC
16	2	2	4	0.1441631466	0.60
17	2	2	6	0.062644951	0.34
18	2	2	8	0.0299784616	0.26
19	2	2	10	0.0297305472	0.29
20	2	3	2	0.0319669321	0.17
21	2	3	3	0.0336108096	0.21
22	2	3	5	0.0249661691	0.21
23	2	3	7	0.0324172936	0.27
24	2	3	12	0.031866461	0.34
25	2	3	17	0.0292190872	0.40
26	2	4	2	0.054452043	0.28
27	2	4	4	0.0227532312	0.20
28	2	4	8	0.0229931753	0.26
29	2	4	12	0.0221513808	0.32
30	2	4	16	0.0136856213	0.35
31	2	5	1	0.0403706655	0.23
32	2	5	2	0.0572797507	0.31
33	2	5	3	0.0226743892	0.20
34	2	5	5	0.015065657	0.20
35	2	5	10	0.0245560985	0.31
36	2	5	15	0.0130372141	0.35
37	2	6	2	0.0233143475	0.20
38	2	6	4	0.0256895386	0.24
39	2	6	9	0.0248862263	0.31
40	2	6	9	0.0164287258	0.28
41	2	6	14	0.0169394407	0.36
42	2	8	2	0.0233340282	0.23
43	2	8	7	0.0171429124	0.29
44	2	8	12	0.0079684565	0.33
45	2	10	2	0.0168836489	0.24
46	2	10	5	0.0187602881	0.29
47	2	10	10	0.0225839484	0.38
48	2	12	3	0.0234465152	0.31
49	2	12	8	0.0112896916	0.34
50	2	15	5	0.0032410533	0.31

## A.2 Función de base radial

**Tabla A.2:** Función de costo para selección de número de neuronas de capa oculta en red neuronal RBF.

Configuración	Neuronas capa oculta	Mse	FC
1	1	0.1793671846	0.7075
2	2	0.1793670654	0.7149995348
3	4	0.1793666929	0.7299980811
4	5	0.0438840576	0.2087623209
5	6	0.0437612794	0.2157831655
6	8	0.0446907841	0.234410659
7	10	0.0438490771	0.2461258056
8	12	0.0449048392	0.2652460324
9	14	0.0437779687	0.2758482974
10	15	0.0438020639	0.2834423315
11	16	0.0439366549	0.2914675876
12	18	0.0437505543	0.3057413097
13	20	0.037734095	0.2972614211
14	30	0.0437556691	0.3957612707
15	40	0.0438087732	0.4709685153



# Apéndice B

## Archivos de redes para biblioteca FANN

### B.1 mlp.net

```
FANN_FLO_2.1
num_layers=3
learning_rate=0.700000
connection_rate=1.000000
network_type=0
learning_momentum=0.000000
training_algorithm=2
train_error_function=1
train_stop_function=0
cascade_output_change_fraction=0.010000
quickprop_decay=-0.000100
quickprop_mu=1.750000
rprop_increase_factor=1.200000
rprop_decrease_factor=0.500000
rprop_delta_min=0.000000
rprop_delta_max=50.000000
rprop_delta_zero=0.100000
cascade_output_stagnation_epochs=12
cascade_candidate_change_fraction=0.010000
cascade_candidate_stagnation_epochs=12
cascade_max_out_epochs=150
cascade_min_out_epochs=50
cascade_max_cand_epochs=150
cascade_min_cand_epochs=50
cascade_num_candidate_groups=2
bit_fail_limit=3.49999994039535522461e-01
cascade_candidate_limit=1.00000000000000000000e+03
```

```

cascade_weight_multiplier=4.00000005960464477539e-01
cascade_activation_functions_count=10
cascade_activation_functions=3 5 7 8 10 11 14 15 16 17
cascade_activation_steepnesses_count=4
cascade_activation_steepnesses=2.50000000000000000000e-01
5.00000000000000000000e-01 7.50000000000000000000e-01 1.00000000000000000000e+00
layer_sizes=4 5 2
scale_included=0
neurons (num_inputs, activation_function, activation_steepness)=
(0, 0, 0.00000000000000000000e+00) (0, 0, 0.00000000000000000000e+00)
(0, 0, 0.00000000000000000000e+00) (0, 0, 0.00000000000000000000e+00)
(4, 5, 5.00000000000000000000e-01) (4, 5, 5.00000000000000000000e-01)
(4, 5, 5.00000000000000000000e-01) (4, 5, 5.00000000000000000000e-01)
(0, 5, 0.00000000000000000000e+00) (5, 0, 5.00000000000000000000e-01)
(0, 0, 0.00000000000000000000e+00)
connections (connected_to_neuron, weight)=(0, 1.43423318862915039062e-01)
(1, -4.59371376037597656250e+00) (2, -7.67581909894943237305e-02)
(3, 7.70345735549926757812e+00) (0, 1.49996472167968750000e+03)
(1, -1.50000000000000000000e+03) (2, 1.35809898376464843750e+00)
(3, 2.71004455566406250000e+02) (0, 1.78982153534889221191e-01)
(1, 9.18564224243164062500e+00) (2, -1.23998872935771942139e-01)
(3, -2.67611980438232421875e-01) (0, 4.16049301624298095703e-01)
(1, -1.03631124496459960938e+01) (2, -1.16443909704685211182e-01)
(3, 2.18302205204963684082e-01) (4, 7.21879184246063232422e-01)
(5, 9.90124762058258056641e-01) (6, 9.57852303981781005859e-01)
(7, -5.37811100482940673828e-01) (8, 9.28429305553436279297e-01)

```

## B.2 rbf.net

```

FANN_FLO_2.1
num_layers=3
learning_rate=0.700000
connection_rate=1.000000
network_type=0
learning_momentum=0.000000
training_algorithm=2
train_error_function=1
train_stop_function=0
cascade_output_change_fraction=0.010000
quickprop_decay=-0.000100
quickprop_mu=1.750000
rprop_increase_factor=1.200000
rprop_decrease_factor=0.500000
rprop_delta_min=0.000000

```







# Índice alfabético

DANN, 14, 27  
deadline, 18, 41, 42

Fast Artificial Neural Network, 4  
función de activación, 10

MLP, 12, 27, 29, 30, 37  
mse, 29

PCANN, 14, 26  
perceptrón, 10  
plasmas fríos, 1, 18  
potencial del plasma, 7

RBF, 14, 27, 29, 37  
retropropagación, 11, 19, 27  
RNN, 12, 26

sigmoidal, 11  
sonda de Langmuir, 7  
Stellarator, 1

Tokamak, 1

validación cruzada, 4, 11, 20  
VI, 21, 23