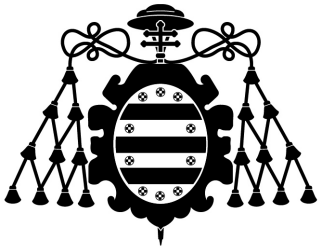
Modelo de Red Neuronal Bayesiana en la Exploración Física de Plasmas de Fusión

Álex Panera Álvarez

Aaron Ho

Roberto Luis Iglesias Pastrana

Susana Montes Rodriguez



Universidad de Oviedo

Trabajo de Fin de Grado

Doble Grado en Física y Matemáticas

9 de junio de 2022

Marco Científico

Esta investigación ha sido apoyada y supervisada por el grupo de investigación *Enabling Research* (**ENR-WP2**) englobado en el consorcio europeo de investigación en fusión nuclear **EUROfusion**.

En especial, el proyecto ha sido supervisado por Aaron Ho, investigador postdoctoral en el instituto de la investigación fundamental de la energía, **DIFFER**.

Agradecimientos

En primer lugar, me gustaría agradecer a mi tutor Aaron por brindarme la oportunidad de realizar este trabajo que para mí ha sido tan especial. Sin su estrecha colaboración no habría sido tan fácil darme cuenta de lo que realmente me apasiona. Del mismo modo al grupo de investigación ENR-WP2 por aportar su visión y ayudarme en esta investigación.
Gracias también a mis dos tutores, Roberto y Susana, que desde el primer momento se mostraron predispuestos a acompañarme en este proyecto.

Por último, pero no menos importante, me encantaría dar las gracias a mi madre, que es quien ha hecho posible que haya llegado hasta aquí. Por supuesto, también gracias a todos aquellos que me han acompañado durante estos largos y a la vez breves cinco años.

Álex Panera Álvarez

Oviedo, 9 de junio de 2022

Resumen

Desde hace más de 70 años, la fusión nuclear es una vía de investigación abierta que tiene el fin de proveer a la humanidad con una fuente de energía limpia e inagotable. En este proyecto, el estado del arte de las técnicas matemáticas de inteligencia artificial realizan una valiosa contribución para acelerar la investigación en fusión nuclear. Como consecuencia, se desarrollan modelos sustitutos que aceleran dos importantes modelos para la experimentación en plasmas de fusión, denominados EPED y EuroPED. A su paso obtienen información y dependencias de las diferentes variables que modelan el plasma. El resultado de este proyecto abrirá nuevas vías de investigación en lo que a la exploración de plasmas de fusión respecta, además de reducir el tiempo de computación de los citados modelos en futuros estudios.

A continuación, se especificará la división del trabajo entre el Grado en Matemáticas y el Grado en Física.

Grado en Física:

1. Introducción.
2. Fundamentos: Secciones 2.1 y 2.2.
3. Metodología.
4. Resultados.
5. Conclusiones.

Grado en Matemáticas:

1. Introducción.

2. Fundamentos: Secciones 2.3, 2.4 y 2.5.

3. Metodología.

4. Resultados.

5. Conclusiones.

Índice general

Capítulo 1

Introducción

En este proyecto nos centraremos en el desarrollo de herramientas matemáticas relacionadas con la inteligencia artificial y su aplicación a problemas concretos en física de plasmas de fusión nuclear. La investigación se encuentra enmarcada en la intersección entre la fusión nuclear por confinamiento magnético y el aprendizaje automático o *machine learning*.

1.1 Motivación

Antes de introducir todas las particularidades técnicas que esta investigación esconde, merece la pena explicar qué es la fusión nuclear y por qué es crucial su desarrollo para el futuro de la especie humana.

1.1.1 ¿Por Qué Fusión Nuclear?

El sueño de la fusión nuclear consiste, a groso modo, en «replicar» las condiciones dentro de una estrella, lo cual hace posible que se liberen inmensas cantidades de energía cuando colisionan dos isótopos del mismo elemento (en el caso terrestre del hidrógeno) y se fusionan para producir núcleos de átomos más pesados (como el helio). Tal y como podemos observar en la ilustración de la figura 1.1.

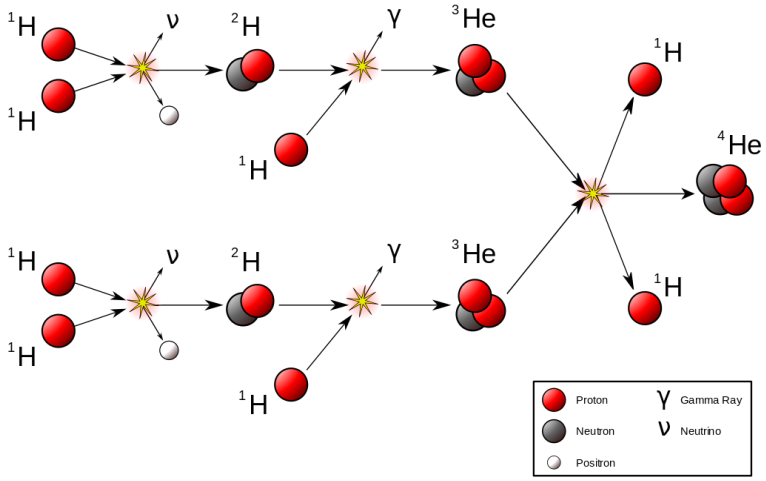


Figura 1.1: Reacción de fusión nuclear llevada a cabo en las estrellas de forma general. Los núcleos de hidrógeno se fusionan para dar lugar, a través de colisiones, a núcleos de helio. Fuente:https://www.euro-fusion.org/fusion/fusion-on-the-sun/.

Sin embargo, el objetivo real para nuestro planeta es un tanto diferente, dada la larga lista de obstáculos que se presentan al tratar de fusionar dos núcleos en uno solo. Por ello una gran variedad de disciplinas de la ciencia y tecnología se verán implicadas en este proceso, tales como física de plasmas, electromagnetismo, física de materiales, física atómica y un largo etcétera.

Centrémonos en cómo «traer el sol a la tierra» puede cambiar nuestras vidas y por qué sin lugar a dudas es uno de los problemas más cruciales y significativos de este siglo, el cual tenemos el deber y la obligación de resolver.

En primer lugar, la especie humana es en su mayoría dependiente de la electricidad, desde que se levanta hasta que se acuesta. Así que la manera en la que producimos esta electricidad va a generar un gran impacto, sobretodo en el medio ambiente. Esto es debido a que si, por ejemplo, nuestra fuente de energía primaria está basada en combustibles fósiles, emitiremos grandes cantidades de dióxido de carbono y otros gases de efecto invernadero [1]. Lo cual a su vez provoca un sinfín de reacciones, tales como elevar la temperatura del planeta atrapando más energía solar en la atmósfera de la necesaria [2]. Las reacciones en cadena comienzan en este punto y se alteran las reservas de agua y los patrones meteorológicos típicos, se producen cambios en las cosechas y por tanto en

la calidad y cantidad de los alimentos, incluso amenaza las comunidades costeras con el incremento del nivel del mar (debido en parte al derretimiento de los casquetes polares causado por el incremento de las temperaturas).

Si bien es cierto que existe una amplia gama de alternativas a los combustibles fósiles, tales como las energías renovables o la energía nuclear de fisión, las cuales son realmente válidas, estas tienen sus ventajas e inconvenientes. Por ejemplo, en lo que se refiere a las energías renovables, estas aprovechan la energía de recursos naturales como la luz solar, el agua y el viento, lo que las hace vulnerables a las condiciones climáticas y por tanto intermitentes en el tiempo. Necesitamos una fuente de energía que sea constante, la cual podría ser la energía nuclear de fisión. Esta toma su energía de la separación de núcleos pesados en núcleos más ligeros. El principal problema de la fisión nuclear son los desechos radiactivos que acarrea, centrándonos en su almacenamiento y la amenaza para la salud que la materia radiactiva supone. Sin tener en cuenta el altamente improbable accidente nuclear que pudiera ocurrir y sus devastadoras consecuencias. A pesar de esto, la energía nuclear es muy segura y su huella de carbono es menor que la de la mayoría de energías renovables, por lo que a corto-medio plazo una mezcla entre energías renovables y fisión nuclear es la opción más adecuada si queremos reducir los problemas medioambientales y de salud por kWh producido.

Sin embargo, a largo plazo es cuando la fusión nuclear comienza a jugar su papel, eso es, una energía constante en el tiempo, con combustible barato y (casi) inagotable, y obviamente sin el impacto ambiental negativo de los combustibles fósiles. La razón por la que se dice que el combustible es interminable y barato es porque en la reacción de fusión usamos, generalmente, deuterio (hidrógeno con un neutrón) y tritio (hidrógeno con dos neutrones). El primero se encuentra fácilmente en el mar, y el segundo, a pesar de tener una vida media de 12,32 años, puede producirse, por ejemplo, mediante la activación neutrónica del Litio-6 en un reactor nuclear.

Los antedichos son dos factores clave para desarrollar la fusión nuclear, podemos reducir significativamente nuestras emisiones de $CO2$ a escala global y alcanzar una energía

Figura 1.2: Imagen ilustrativa de la mezcla de energías renovables y nucleares. Fuente: https://www.dianuke.org/nuclear-or-renewable-what-fights-climate-change-better-read-benjamin-k-sovacools-analysi

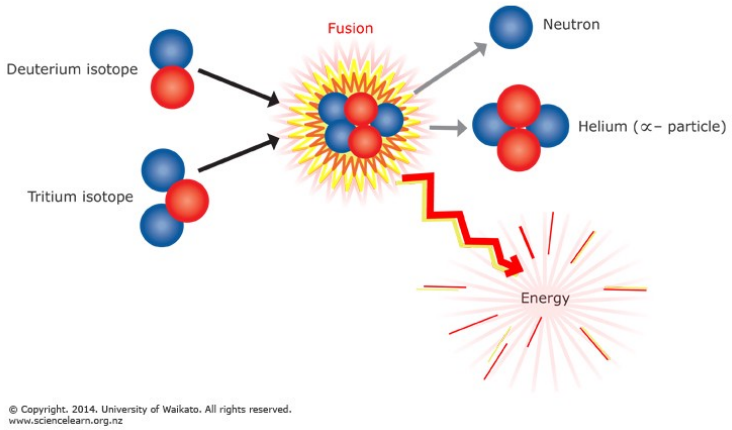


Figura 1.3: Reacción más común para conseguir la fusión nuclear en nuestro planeta. Utilizando deuterio y tritio. Fuente: https://studentsxstudents.com/plasmas-and-nuclear-fusion-7cb53735db0f

ilimitada y limpia que cambiará nuestra proyección ambiental hacia el futuro. Como es de suponer, queda un largo camino por recorrer, y por eso hablamos de largo plazo. Desde la década de 1930, el ser humano ha estado buscando hacer realidad el sueño de la fusión nuclear, y una gran lista de investigadores nos han acercado hasta donde nos encontramos hoy en día. Actualmente existe un gran proyecto público de reactor de fusión nuclear, llamado ITER (*International Thermonuclear Experimental Reactor*), construido en Cadarache, Francia. Prometen lograr la fusión nuclear como fuente de energía factible

en 2050, pero una docena de empresas privadas pretenden alcanzar ese objetivo con anterioridad.

1.2 Problema a resolver

Tras entender la magnitud del campo y su relevancia, podemos presentar uno de los muchos inconvenientes que se encuentran a su paso las investigaciones y proyectos relacionados con la fusión nuclear.

Como será detallado en el siguiente capítulo, las reacciones de fusión nuclear ocurren en estado de plasma, el cual podemos definir, sin entrar en detalles, como un gas ionizado. Este plasma se convierte en un sistema muy complejo, el cual es importante modelizar y conocer su comportamiento. Con este propósito se idean códigos de simulación y modelos de predicción sobre diferentes parámetros del propio plasma, tales como temperatura, densidad y un largo etcétera que pronto detallaremos. El principal problema de estas herramientas, tan útiles para tratar los plasmas de fusión, es su peso computacional y por tanto su velocidad. Es decir, son incompatibles con la modelización en tiempo real, lo cual resulta de gran valor en gran cantidad de experimentos y, por supuesto, en el desarrollo de la fusión nuclear como fuente de energía.

Particularmente, este proyecto se centra en los llamados, modelos de pedestal del plasma que, en lineas generales, predicen parámetros del plasma centrándose en el salto brusco o pedestal que se produce cuando nos alejamos del núcleo del mismo.

1.3 Objetivos y Contribución

El objetivo principal del trabajo pues, es solventar el problema de la velocidad de los modelos de pedestal de plasma a la hora de predecir parámetros. Los principales modelos sobre los que se trabajará son los modelos EPED y EuroPED descritos con gran detalle en los apartados 2.2.1 y 2.2.2 respectivamente.

El método para llevar a cabo el aceleramiento de estos modelos está íntimamente relacionado con la inteligencia artificial, en concreto, se trata de una red neuronal Bayesiana con contraste de ruido. Este método se explicará con detalle en la segunda mitad del siguiente capítulo.

El resultado no es otra cosa que un modelo sustituto que replica el comportamiento del modelo original, solo que su peso y tiempo de computación se ven drásticamente reducidos. Una vez desarrollado este modelo, sus aplicaciones no se limitan a la suplantación y la aceleración del modelo original, sino que dada la simplificación realizada, podemos apreciar las características y dependencias del modelo original de forma mucho más clara. Lo cual se traduce en las características y dependencias del propio plasma de fusión.

Sin embargo, nuestro método para conseguir el modelo sustituto tiene además ciertas peculiaridades. Por tratarse de un método Bayesiano un tanto especial, manejaremos el concepto de *prior*. Esto nos permitirá especificar cierta información antes del entrenamiento de la red, información relacionada, por ejemplo, con la física subyacente del sistema. Se trata de un valor añadido de gran relevancia a un modelo que de otra manera estaría puramente cimentado sobre datos.

Otra peculiaridad de gran relevancia de nuestro modelo sustituto es que ofrece información sobre la incertidumbre de sus predicciones, segmentada en dos tipos, el error debido al propio modelo y el debido al ruido de los datos que se le introducen. De este modo, en sus múltiples aplicaciones podemos conocer la precisión que estamos manejando.

En conclusión, los modelos sustitutos nos abren un amplio abanico en la interpretación y aplicación de los modelos de plasmas. Incluso en la interpretación de los parámetros y la física subyacente detrás de la propia experimentación.

De esta forma, a través del objetivo principal del proyecto quedan abiertas nuevas vías

de investigación que merecen ser exploradas en el futuro.

1.4 Sumario

Resumidamente, este proyecto se centra en desarrollar un modelo sustituto para los modelos de plasma de pedestal EPED y EuroPED, mediante el método de red neuronal bayesiana con contraste de ruido (BNN-NCP). Además se realiza un análisis detallado sobre el comportamiento del modelo sustituto y la información que ofrece, tanto en los datos provenientes de los citados modelos, como en los datos experimentales. Todos los modelos se cimentan en los datos obtenidos del Tokamak JET en el experimento JET-ILW y en la aplicación de los modelos sobre los mismos. El conjunto de datos se describe en la sección 3.3.

En el capitulo 2, se describen los fundamentos necesarios para entender y situar completamente los avances científicos llevados a cabo en este proyecto. En el capítulo 3, se presenta y detalla el método construido y los datos con los que se alimentará la red. En el capítulo 4, se describen y discuten los resultados de la implementación del método que da lugar a los modelos sustitutos, además se analizan sus consecuencias inmediatas. En el capítulo 5, se resumen y discuten los avances y sus consecuencias, además de las vías de investigación que quedan abiertas.

Capítulo 2

Fundamentos

2.1 Fusión Nuclear

El principio de la energía de fusión consiste en fusionar dos núcleos más ligeros en uno más pesado, liberando energía en el proceso. Actualmente, la reacción más prometedora es la fusión del deuterio y el tritio en helio, liberando un neutrón y energía en el proceso [3]. Los beneficios de esta elección son varios, pero el más relevante es la sección eficaz de reacción relativamente alta a temperaturas relativamente bajas en comparación con otras reacciones. El deuterio existe en el agua de mar en cantidades suficientes para satisfacer nuestras necesidades energéticas en el futuro previsible. Sin embargo, el tritio existe muy escasamente en la naturaleza, ya que es un isótopo radiactivo que se desintegra. Debido a esto, uno de los requisitos para la fusión como futura fuente de energía es encontrar una forma eficiente de producir tritio. El método propuesto más importante para esto se lleva a cabo a través de la «reproducción» de tritio en una región de manto alrededor del reactor. Esta región de cobertura permitiría que el exceso de neutrones producidos en la reacción de fusión interactúe con el litio, fisionándolo y, por lo tanto, creando tritio de forma autosuficiente, este manto se denomina *tritium breeding blanket*, y será una parte esencial de futuros reactores de fusión.

Lograr la fusión no es una tarea sencilla, ya que los núcleos atómicos están cargados positivamente, lo que genera una fuerza de repulsión entre ellos. Los dos núcleos necesi-

tan velocidades muy altas para vencer esta fuerza y la energía requerida se llama barrera de Coulomb. Una forma de lograrlo es a través de la fusión termonuclear, que implica calentar un grupo de núcleos hasta que sus velocidades térmicas sean lo suficientemente altas como para superar la barrera de Coulomb. Esto requiere temperaturas extremadamente altas, del orden de los 100 millones de grados Kelvin. Actualmente, en muchas instalaciones de investigación de fusión, las partículas se encuentran en forma de plasma que debe ser confinado. El confinamiento es importante por muchas razones. Una es mantener el plasma en un volumen pequeño para que la densidad y la presión se mantengan altas. Otra razón es proteger cualquier material de pared de las temperaturas extremas del plasma y del mismo modo para proteger el plasma de interactuar con cualquier material de la pared y, por lo tanto, que se enfríe y se contamine.

2.1.1 Confinamiento magnético y Tokamak

Dado que un plasma es una colección de partículas cargadas, este puede estar confinado por un campo magnético. Mediante la ecuación $\mathbf{F} = q\mathbf{E} + q\mathbf{v} \times \mathbf{B}$ obtenemos la fuerza de Lorentz $\mathbf{F}$ que experimenta una partícula con carga q y velocidad $\mathbf{v}$ en un campo eléctrico y magnético $\mathbf{E}$ y $\mathbf{B}$ respectivamente. Esta fuerza hace que las partículas cargadas puedan moverse libremente a lo largo de las líneas del campo magnético. Sin embargo, si intentan moverse perpendicularmente a las líneas del campo magnético, se ejerce una fuerza que cambia la dirección de la trayectoria de las partículas.

Asumiendo la ausencia de campo eléctrico, la fuerza ejercida siempre es perpendicular a la trayectoria de las partículas, por lo que no se añade energía a dicha partícula. Si también asumimos un campo magnético uniforme, la magnitud de la fuerza será constante y se puede equiparar con una fuerza centrípeta que obliga a la partícula a seguir las líneas del campo magnético en una trayectoria helicoidal, ya que no puede moverse libremente en el plano perpendicular a las líneas de campo magnético. Esto básicamente limita a la partícula a moverse en una superficie cilíndrica cuyo radio está determinado por la fuerza del campo magnético, la masa de la partícula y la componente de la velocidad

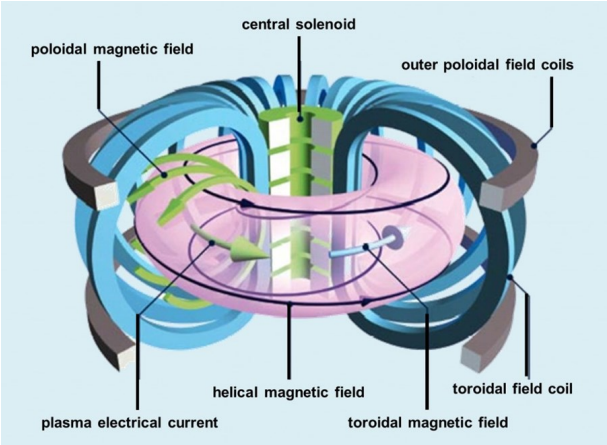


Figura 2.1: Imagen esquemática de un Tokamak. Fuente: ITER https://www.iter.org/newsline/-/3037

perpendicular a las líneas de campo.

En un Tokamak, las lineas de campo magnético son curvadas formando un toro, de esta forma las partículas cargadas son forzadas a seguir una trayectoria circular. Esto se muestra en la figura 2.1, donde el campo magnético toroidal se crea mediante unas bobinas toroidales que se anidan sucesivamente. Anteriormente asumíamos que el campo magnético era uniforme y no existía ningún campo eléctrico, sin embargo, dado que las lineas de campo se curvan a lo largo del toro, esto provoca que este campo magnético ya no sea uniforme. De esta forma, en la zona más interior del toro, el campo magnético se vuelve ligeramente más intenso que en la zona más exterior del mismo, lo cual origina las siglas HFS (*High Field Side*) y LFS (*Low Field Side*) respectivamente. Este gradiente de intensidad de campo magnético origina que la partícula que orbita en una trayectoria circular a lo largo del toro se desvíe en función de su carga eléctrica. Y por tanto se produce una separación de cargas, creando un campo eléctrico que deriva en perdida del confinamiento del plasma. Una manera de resolver este problema es conectando la parte superior e inferior del plasma, lo cual equivale a añadir una corriente al plasma, esto induce un campo magnético poloidal (ambas representadas en la figura 2.1 con flechas verdes). La corriente del plasma se inicia generalmente por un solenoide interno, que también se muestra en la figura. La adición del campo magnético poloidal resulta en un campo magnético helicoidal que compensa adecuadamente las desviaciones anteriormente mencionadas, de esta forma se consigue confinar el plasma correctamente. A lo largo

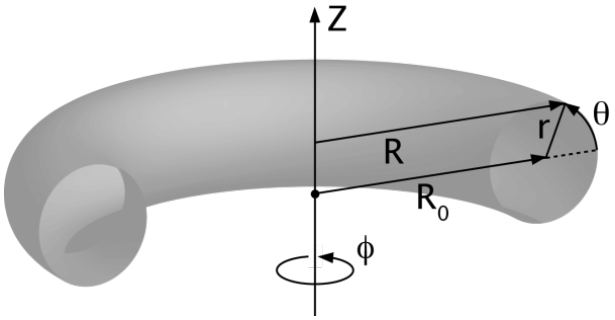


Figura 2.2: Coordenadas toroidales. Fuente: CIEMAT. http://fusionwiki.ciemat.es/wiki/Toroidal_coordinates

del proyecto nos referimos a la corriente eléctrica del plasma como I_p, y aunque incluye la componente toroidal y la poloidal generalmente se toma como la componente tororidal, ya que esta componente domina respecto de la otra, esta corriente I_p es efectivamente la que creo el campo magnético poloidal. Al campo magnético toroidal lo denominaremos B_t.

Dada la geometría toroidal del tokamak y sus simetrías, el plasma se describe usando coordenadas toroidales, tal y como se muestra en la figura 2.2. De esta forma cualquier punto dentro del plasma puede ser descrito mediante el ángulo toroidal Φ, el ángulo poloidal θ y el radio menor r. Sin embargo, la coordenada radial puede ser descrita por una cantidad diferente que garantiza ciertas simetrías en el ángulo poloidal.

En el plasma de un tokamak existen superficies de flujo magnético donde ciertas cantidades son preservadas. Estas superficies de flujo magnético son superficies cerradas en el plasma donde el producto escalar del campo magnético y la normal a la superficie es cero en todos los puntos. En un tokamak, estas superficies son toros anidados. Una forma de referirse a estas superficies es mediante el flujo magnético que pasa a través de ellas, desde el eje magnético (en el medio del plasma) hasta la propia superficie de flujo. Este se llama flujo poloidal ψ. Las superficies de flujo nos dan una nueva forma de describir la coordenada radial menor para un plasma donde se garantiza que ciertas cantidades son constantes en θ para un ψ dado [4].

Un factor importante a tener en cuenta en el rendimiento del plasma son las impurezas

que contiene. Estas acostumbran a ser descritas por la carga efectiva de los iones, que se define en la ecuación 2.1, donde n_i es el la densidad de iones con carga Z_i. Generalmente, en los plasmas de fusión las temperaturas son tan altas que la mayoría de los átomos están completamente ionizados. Esto implica que la densidad de electrones se puede aproximar por $n_E = \sum_k n_k Z_k$.

$$Z_{eff} = \frac{\sum_k n_k Z_i^2}{\sum_k n_k Z_i} = \frac{\sum_k n_k Z_i^2}{n_e} \tag{2.1}$$

Las impurezas en el plasma pueden provenir de diferentes fuentes: los propios productos de la fusión, la pared de la cámara del tokamak o pueden ser introducidos en el plasma intencionadamente.

Si todas las superficies de flujo magnético fueran cerradas, las impurezas se acumularían en el interior del plasma afectando severamente su rendimiento. Por lo cual, a partir de un determinado punto radial las superficies de flujo son por lo general abiertas, de esta manera se crea un canal a través del cual las partículas pueden abandonar el plasma. Denominamos *Separatrix* a la última superficie de flujo cerrado, esta forma un punto característico denominado *X-point*. Todo esto se observa en la figura 2.3 que nos muestra la sección poloidal del plasma con tres superficies de flujo: una cerrada, la *separatrix* y una abierta. La trayectoria abierta en el plasma confinado se dirige hacia las placas de desvío (en inglés *divertor plates*). Estás placas están diseñadas para una mayor tolerancia al intenso bombardeo de partículas a altas temperaturas si las comparamos con otros componentes que se enfrentan al plasma tales como las paredes internas del tokamak. La normalización del flujo poloidal es tal que el eje magnético (centro del plasma) es 0 y la *separatrix* es 1. Debido al método de cálculo del flujo poloidal a partir del campo magnético, no se garantiza automáticamente que el eje magnético tenga un valor de 0 de forma natural. Por lo tanto, la ecuación del flujo poloidal normalizado corresponde con la ecuacion 2.2, donde ψ_{sep} es el flujo poloidal en la *separatrix* y ψ_{eje} en el eje magnético.

$$\psi_N = \frac{\psi - \psi_{eje}}{\psi_{sep} - \psi_{eje}} \tag{2.2}$$

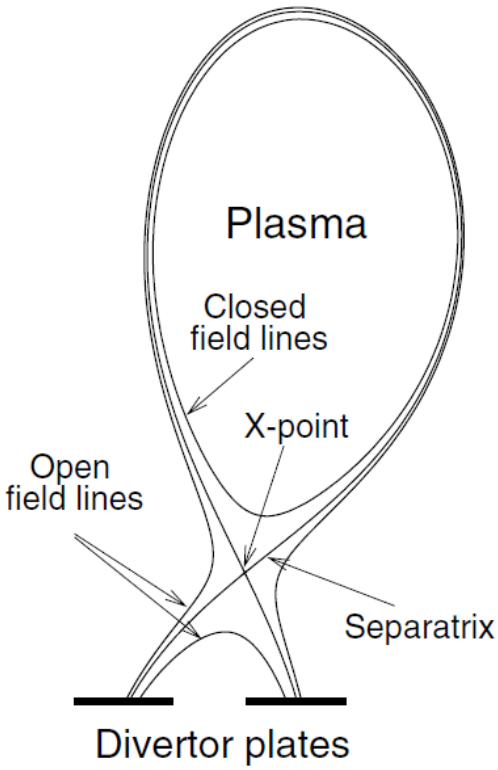


Figura 2.3: Vista esquemática de la sección poloidal de las superficies de flujo. Fuente: [5]

La forma del plasma de un tokamak a menudo se caracteriza por varios parámetros, en las siguientes definiciones R y z se refieren a la figura 2.2. Uno de los parámetros de forma más fundamentales es el radio mayor que viene dado por la ecuación 2.3, donde R_{max} y R_{min} son el valor máximo y mínimo de R a lo largo de la *separatrix*.

$$R_0 = (R_{max} + R_{min})/2 \tag{2.3}$$

Otro es el radio menor del tokamak que se calcula mediante la ecuación 2.4.

$$a = (R_{max} - R_{min})/2 \tag{2.4}$$

Además de estos dos parámetros, la triangularidad será también relevante debido a su influencia en el comportamiento del plasma. La triangularidad se puede dividir en la triangularidad superior e inferior, donde la triangularidad superior viene dada por la ecuación

2.5 siendo $R_{z_{max}}$ el valor de R en el z más alto a lo largo de la *separatrix*.

$$\delta_u = (R_0 - R_{z_{max}})/a \tag{2.5}$$

La triangularidad inferior δ_l se define de la misma manera pero usando el valor de R en la z más baja a lo largo de la *separatrix* en su lugar. La triangularidad total de un plasma es por tanto el promedio de la triangularidad superior e inferior.

Hay varios parámetros diferentes que se pueden utilizar para evaluar el rendimiento de un tokamak. Uno de esos parámetros es la beta del plasma, que es la relación entre la presión del plasma y la presión magnética. A lo largo del proyecto haremos uso de su componente poloidal β_p, es decir la correspondiente a la coordenada del ángulo θ en la figura 2.2, la cual se define en la ecuación 2.6 donde $< p >$ es el valor medio de la presión según las superficies de flujo, $< B_p >$ es el valor medio de la coordenada poloidal del campo magnético y μ_0 es la permeabilidad magnética en el vacío. Vemos como la presión se puede sustituir mediante la densidad electrónica n_e y la temperatura electrónica del plasma T_e. Estás serán variables de gran importancia en el proyecto. Notar que existe un valor de β_p por cada superficie de flujo.

$$\beta_p = \frac{< p >}{< B_p >^2 /2\mu_0} = \frac{< n_e T_e >}{< B_p >^2 /2\mu_0} \tag{2.6}$$

También definimos la beta plasmática total cómo en la ecuación 2.7, donde $< p >_V$ es el valor medio de la presión según el volumen y B_0 es el campo magnético en el eje magnético, donde se asume que la coordenada poloidal en ese punto B_p es pequeña y despreciable. Notar que, en este caso existe un único valor de β_n global para cada descarga del plasma.

$$\beta_n = \frac{< p >_V}{B_0^2/2\mu_0} \tag{2.7}$$

La β del plasma es básicamente una medida de como de efectiva es una configuración magnética confinando el plasma para un determinado campo magnético. Como β está determinada por cantidades locales de presión y campo magnético, esta puede variar dentro del plasma. Otro parámetro importante en la evaluación del rendimiento de un tokamak

es el tiempo de confinamiento energético τ_E. El tiempo de confinamiento energético es una medida del tiempo que tarda un plasma en ceder energía a su entorno. Como tal, es una medida directa de la propiedades de confinamiento del dispositivo. Si el tiempo de confinamiento de energía se duplica esto implica que la configuración es el doble de buena para confinar la energía del plasma.

2.1.2 Pedestal y *H-mode*

Magnetohidrodinámica (MHD)

Dado que el plasma en un tokamak es un sistema con gran complejidad y una densidad de partículas del orden de $10^{20}\,m^{-3}$, se necesita de modelos simplificados para describir su comportamiento. De esta manera, el modelo magnetohidrodinámico (MHD) ha demostrado ser eficiente describiendo inestabilidades del plasma a gran escala. El modelo entiende el plasma como un fluido eléctricamente conductor, y se deriva de las ecuaciones de Navier-Stokes y Maxwell, describiendo el plasma con cantidades macroscópicas como temperatura, presión y densidad.

Para que el plasma de fusión sea estable ha de estar en equilibrio según el modelo MHD, lo que significa que no haya cambios en sus propiedades macroscópicas a lo largo del tiempo sin influencia del exterior. Sin embargo, esto no corresponde con un sistema real, donde aparecen perturbaciones, de esta manera si el equilibrio es estable hacia esta perturbación, el sistema volverá al equilibrio, si no lo es, la perturbación aumentará. Estas inestabilidades son básicamente deformaciones del toro en el que está confinado el plasma, que se pueden describir separando en coordenadas como en la figura 2.2 y añadiendo el tiempo. Tomando transformadas espaciales de Fourier en 2D en las coordenadas toroidales y poloidales aparecen los números m y n referidos a los modos poloidales y toroidales respectivamente, tal y como se aprecia en la ecuación 2.8.

$$\xi(r,\theta,\phi,t) = \sum_m \sum_n \xi_{m,n}(r,t) e^{i(m\theta - n\phi)} e^{\gamma_{m,n} t} \tag{2.8}$$

En la ecuación 2.8, ξ representa la magnitud de la perturbación y γ la tasa de crecimiento. Esta separación en modos es crucial para describir el comportamiento del plasma con respecto a la estabilidad. Ahora definiremos un nuevo parámetro, el factor de seguridad q, para ello debemos recordar que en un reactor de fusión toroidal (como los tokamak), los campos magnéticos que confinan el plasma se forman en forma helicoidal, enrollándose alrededor del interior del reactor. El factor de seguridad q es la relación entre las veces que una determinada línea de campo magnético recorre el «camino largo» (toroidal) de una zona de confinamiento toroidal y el «camino corto» (poloidal). Se puede tomar el factor de seguridad en un tokamak aproximadamente como $q = \frac{aB_t}{R_0 B_p}$. La teoría MHD exige para la estabilidad del plasma $q > 2$ en la separatrix [6]. Aunque esencialmente $q = m/n$ (relación entre modos poloidales y toroidales) y este varía en función del radio menor, a. Así, cuando m/n es racional (m y n son enteros respectivamente), estos modos MHD pueden estar presentes en ese radio menor a. En el borde del plasma, q varía típicamente en un orden de magnitud o más en ese pequeño espacio. Esto proporciona un gran número de superficies resonantes para que los modos MHD aparezcan y sean inestables. La caracterización de estos modos y sus interacciones sigue siendo un campo de investigación activo, en el que la descripción *peeling-ballooning* (se explica a continuación) ha sido el intento más exitoso, sin embargo experimentalmente no describe adecuadamente todos los pedestales (este concepto será explicado con detalle en el siguiente apartado).

Las inestabilidades predichas por el modelo MHD se separan según diferentes categorías. Existe una división entre modos de resistencia y modos ideales, los primeros se relacionan con la resistencia finita del plasma, mientras que los segundos aparecen incluso en condiciones de conducción perfecta. Otra división se produce entre las fuerzas que provocan esa inestabilidad: modos causados por la presión y modos causados por la corriente. Los llamados *peeling modes* son provocados por la corriente eléctrica, y los *ballooning modes* por la presión. Estos modos serán de gran importancia para describir la estabilidad del plasma, pudiendo acoplarse en *peeling-ballooning modes*.

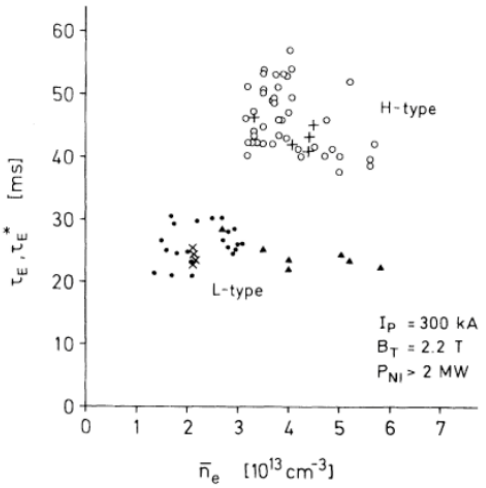


Figura 2.4: Tiempo de confinamiento de la energía global frente a densidad media para el limitador toroidal (triángulos) y descargar del divertor (otros símbolos). τ_E (signo de adición y cruces) se deduce de los perfiles térmicos y τ_E^* (circunferencias, círculos y triángulos) se determina a partir de la medida diamagnética de $\beta_{p\perp}$. Fuente: [8]

H-mode y ELM

[7] En 1982, un nuevo modo de operación se descubrió en el tokamak ASDEX [8]. El nuevo modo, denominado modo-H (modo de alto confinamiento) en oposición al modo-L (bajo confinamiento), fue alcanzado mediante el incremento de la potencia inyectada al plasma pasado un cierto valor límite, en ese punto el tiempo de confinamiento energético τ_E incluso se hacía más del doble en algunos casos. Debido a esta característica, el modo-H es el régimen deseado en la mayoría de los tokamak. En la figura 2.4 podemos observar la separación de ambos modos L y H, obteniendo el modo H un tiempo de confinamiento de la energía sensiblemente mayor. Tras un análisis más detallado, se encontró que el nuevo modo de operación involucraba la formación de un pedestal en las regiones exteriores del plasma donde la densidad del plasma y la temperatura cambiaban rápidamente hacia una menor densidad y temperatura propias de la región fuera de las superficies de flujo cerrado. En la figura 2.5 observamos la formación del pedestal en el modo-H, frente al modo L. La forma del pedestal se describe habitualmente mediante su anchura y altura, que se suelen estimar a partir de los datos experimentales con un ajuste mtanh (tangente hiperbólica modificada). En este ajuste se profundizará en el apartado 2.1.2.

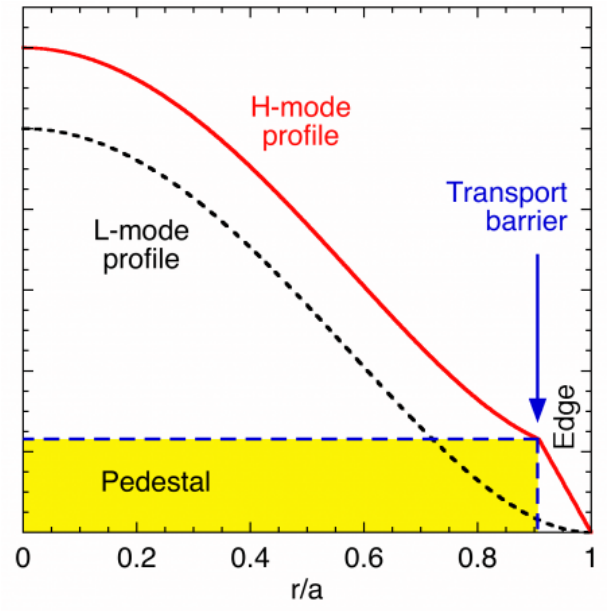


Figura 2.5: Representación esquemática de los perfiles de modos L y H. Fuente: [9]

De esta manera, el plasma en el modo de alto confinamiento (modo H) dentro de la última superficie de flujo cerrada puede ser dividido en dos regiones distintas: el núcleo, que cubre la mayoría del volumen del plasma y se caracteriza por el transporte turbulento de calor y partículas y gradientes relativamente poco pronunciados, y el pedestal, que es una región estrecha cercana al borde del plasma donde las turbulencias a escala iónica son mayormente suprimidas y por lo tanto, encontramos gradientes de presión muy pronunciados. Los perfiles de densidad y de temperatura del núcleo están regulados principalmente por procesos de transporte, mientras que en el pedestal, las inestabilidades MHD que causan relajaciones intermitentes del gradiente de presión, juegan un papel muy importante limitando dicha presión.

Es interesante comentar que las condiciones exactas para la formación del modo H son un campo de investigación activo. Existe una ley de escala empírica para la potencia de entrada necesaria en el tokamak para entrar en el modo H, pero todavía no hay una teoría fundamental detrás del mecanismo. Es decir, el modo H es puramente empírico.

Si bien las condiciones de confinamiento del modo H son generalmente deseables, habría un precio a pagar: la aparición de explosiones de luz D_α (L_α, $\lambda = 1215.67\,\text{Å}$) en el

borde del plasma, lo que indica que algunos estallidos transitorios de gas frío están interactuando con el plasma caliente que se expulsa desde el interior del flujo cerrado. Las señales magnéticas también confirmaron la correlación de estas ráfagas con perturbaciones de números modales m altos (en el rango de 8 a 12). Además, hubo reducciones en el densidad del centro del plasma asociada con estos estallidos. Esta actividad magnética ha sido etiquetada como «Modos localizados en el borde» o ELM. De hecho el diseño del proyecto ITER tiene en cuenta estos sucesos, siendo los ELM un gran inconveniente ya que además de disminuir el rendimiento del confinamiento del plasma pueden producir daños en las paredes del dispositivo.

La formación del pedestal es ocasionalmente acompañada por estas descargas periódicas (ELM) [10]. Por lo que estos sucesos parecen ser el principal factor limitante en la altura del pedesta. Generalmente la anchura y el gradiente del pedestal crecen hasta que un ELM aparece por lo que se produce el colapso del pedestal. Tras esto el pedestal se estabiliza de nuevo incrementando su anchura y gradiente otra vez, hasta que otro ELM aparece. Esto deriva en un comportamiento cíclico. Sin embargo, en algunos regímenes de operación o configuraciones estos ELM no se producen, por ejemplo en el *quiescent H-mode* [11] donde otros efectos físicos limitan el pedestal en su lugar.

Estos gradientes tan pronunciados en las capas más externas del plasma se cree que excitan las inestabilidades de MHD mediante los modos provocados por la presión (*ballooning*) y los provocados por la corriente (*peeling*). Simulaciones computacionales del plasma usando códigos basados en modos resistivos MHD respaldan esta afirmación [12; 13].

Estabilidad del Pedestal

Las llamadas ELM de tipo I, desde su descubrimiento, están estrechamente relacionadas con los modos *ballooning, peeling* y su acoplamiento, por lo que este modelo de estabilidad ideal MHD describe bien este tipo de ELM. De esta manera grandes corrientes en el borde desestabilizan los modos *peeling* de baja n y altos gradientes en la presión desestabilizan modos *ballooning* con alta n. Por ello, mediante este modelo hallamos un dominio de estabilidad y otro de inestabilidad separados por una frontera, la cual llama-

remos límite *peeling-balloning* (PB) de estabilidad. Otros tipos de ELM tales como los de tipo III son observados habitualmente mucho antes de cruzar este límite PB por lo que no están bien descritos por el modelo, debido a que se trata de un modelo ideal donde no se tiene en cuenta la resistividad que desestabiliza el plasma y provoca estos últimos ELM.

El modelo PB se ha implementado en varios códigos de simulación tales como ELITE [14] y MISHKA [15] para desarrollar análisis de estabilidad del pedestal y localizar los ELMs. El mencionado límite de estabilidad PB utilizado en el modelo y los códigos se puede estudiar mediante el espacio de fases $j-\alpha$ donde j representa la densidad de corriente en el borde y α es el gradiente de presión normalizada, definida como en [16].

$$\alpha = -\frac{2\partial_\psi V}{(2\pi)^2}\left(\frac{V}{2\pi^2 R_0}\right)^{\frac{1}{2}}\mu_0\partial_\psi p \tag{2.9}$$

En la ecuación 2.9 se define la cantidad α, donde V representa el volumen del plasma, μ_0 la permeabilidad magnética en el vacío y p la presión total.

En la figura 2.6 observamos el límite de estabilidad PB en el espacio de fases $j-\alpha$. De esta manera, esperamos que el pedestal del plasma evolucione dentro de la región de estabilidad, trazando una determinada trayectoria hasta llegar al límite de estabilidad donde se activaría un ELM. Este ELM (*Edge Localized Mode*) provoca un colapso del pedestal, reduciendo el gradiente de presión y la densidad de corriente j, tal y como observamos en los puntos 1, 2 y 3 de la figura 2.6(a). De este modo el pedestal se estabiliza de nuevo y evoluciona hasta alcanzar de nuevo el límite de estabilidad, creando un patrón recurrente de modos ELMs.

Lo que realmente nos indica el límite de estabilidad es que al menos un modo de perturbación se ha vuelto inestable, sin embargo, diferente regiones de este límite podrían ser desestabilizadas por distintos efectos. Es decir, el modo que se vuelve inestable puede ser diferente a lo largo del límite de estabilidad, de este modo en la figura 2.6(b) podemos ver que tipo de modo se vuelve inestable según la zona donde nuestro punto operacional (configuración momentánea del plasma) interseca con el límite. Las flechas en la figura

2.6(b) muestran la dirección de número modal toroidal (n) creciente del modo de perturbación que se vuelve inestable. La forma del límite PB no es única, sino que depende de diferentes parámetros del plasma tales como triangularidad, campo magnético toroidal, corriente del plasma, etc.

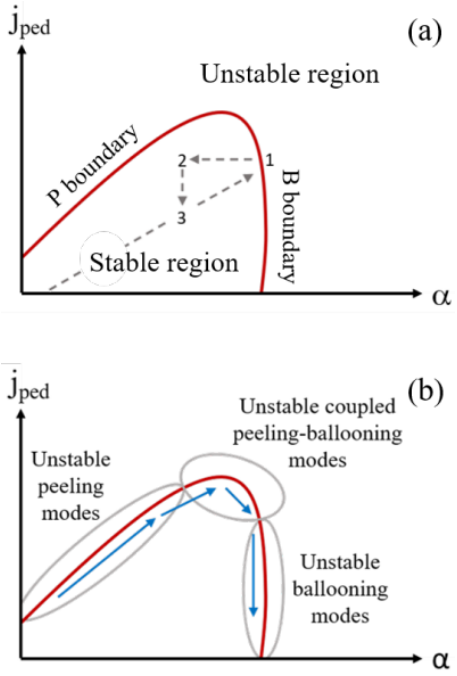


Figura 2.6: Representación del esquema de estabilidad PB con (a) una descripción de como un ELM ocurre y evoluciona y (b) el tipo de modos que desestabilizan el pedestal en cada zona del límite PB, con flechas indicando la dirección de crecimiento del modo inestable n. Fuente: [17]

En la región del pedestal del plasma, la densidad de corriente j está dominada por la llamada corriente *bootstrap*. Si la presión del plasma varía a lo largo del radio del tokamak (como ocurre en el caso del pedestal), surgirá espontáneamente una corriente autogenerada dentro del plasma, debido a las colisiones entre las partículas atrapadas y las partículas que circulan, este fenómeno recibe el nombre de corriente *bootstrap*. De manera que esta corriente está provocada por gradientes de presión en vez de por campos eléctricos, de manera que es mucho más intensa en la zona del pedestal comparada con el resto del plasma.

Además la corriente *bootstrap* depende fuertemente de la tasa de colisiones del plasma [18]. La colisionalidad es un parámetro adimensional que nos ofrece una medida del ratio entre la frecuencia de colisiones del plasma y la frecuencia del movimiento de las par-

tículas del plasma. De esta forma, esta dependencia produce que en un plasma de alta colisionalidad la corriente *bootstrap* desaparece, por lo que tendríamos un pedestal limitado por los modos *ballooning* con número toroidal n alto. Tal y como apreciamos en la figura 2.6(b), dado que j_{ped} disminuiría drásticamente, entonces la estabilidad solo se vería comprometida por los modos *ballooning* con n alta.
Consecuentemente, en un plasma de baja colisionalidad esta corriente no desaparecería, de esta forma para cada α tendríamos una mayor densidad de corriente j, resultando, tal y como se aprecia en la figura 2.6(b), en un pedestal mayormente limitado por modos *peeling* con menor número toroidal n desestabilizando el plasma [14].

La expresión para la colisionalidad electrón-electrón $\nu*_{ee}$ según [19] se encuentra en la ecuación 2.10, donde $\varepsilon = \frac{a}{R0}$ es el radio inverso, q_{95} es el factor de seguridad en $\psi_N = 0.95$, $n_{e,ped}$ y $T_{e,ped}$ la densidad electrónica y la temperatura en lo alto del pedestal respectivamente, y $ln(\Lambda)$ es el *logaritmo de Coulumb*. Vemos en esta expresión que $\nu*_{ee}$ es proporcional a n_e/T_e^2.

$$\nu*_{ee,ped} = \frac{R_0 q_{95}}{\varepsilon^{\frac{2}{3}} \lambda_{ee}} \tag{2.10}$$

$$\lambda_{ee} = 1.47 \cdot 10^{23} \frac{T_{e,ped}^2}{n_{e,ped} ln(\Lambda)}$$

La correlación de la corriente *bootstrap* con la colisionalidad lleva consigo una fuerte dependencia de la colisionalidad en la trayectoria en el espacio de fases $j - \alpha$, y por tanto también en el punto de intersección con el límite PB. Además, ha sido demostrado que la colisionalidad tiene un gran efecto en la precisión de varios modelos de corriente *bootstrap* [19] así como en el tamaño de las descargas de ELM cuando se alcanza el límite de estabilidad [20].

Ajuste de pedestal *mtanh*

Tal y como se ha comentado en la definición de pedestal del plasma, la forma de este se puede definir mediante su altura, anchura y gradiente. Estos cálculos son importantes debido a que la calidad del confinamiento del modo H está fuertemente relacionada con la altura del pedestal para la presión en el modo H. Además el gradiente de la presión está limitado por los modos del modelo MHD, en particular por los modos *ballooning* [21].

Una manera de estimar estas cantidades a partir de los datos experimentales es mediante un ajuste a la función mtanh, o lo que es lo mismo, tangente hiperbólica modificada. De esta forma ajustamos los perfiles de temperatura, presión y densidad en el borde. Anteriormente, este ajuste se realizaba a una función basada en la propia función tanh. Después se le añadió un término lineal para proporcionar una buena conexión con los perfiles del núcleo [22]. Esta función tenía una discontinuidad en la primera derivada en la unión entre los términos lineales y tanh. Para solventar esto se ha implementado una función ligeramente modificada, llamada mtanh, que tiene una primera derivada espacial continua y proporciona casi exactamente el mismo ajuste que la función original. Como muestra la figura 2.7, la función mtanh se ha implementado expandiendo la función tanh en términos de sus exponenciales y luego multiplicando la exponencial apropiada por $(1+\alpha z)$, donde α es una constante que permite un perfil lineal ascendente en el núcleo. Los valores del punto de simetría, el valor del pedestal, la anchura de la barrera, el gradiente en el punto de simetría y todas las demás cantidades de interés son prácticamente idénticas para las dos funciones de ajuste.

$$mtanh(r) = \frac{h}{2}\left[\frac{(1+sx)e^{x} - e^{-x}}{e^{x} + e^{-x}} + 1\right] \tag{2.11}$$

$$x = \frac{2(p_{pos} - r)}{\Delta}$$

En la ecuación 2.11 vemos la expresión desarrollada para la función *mtanh*, donde h representa la altura del pedestal, Δ la anchura del pedestal, p_{pos} es la posición del pedestal dada por el punto medio y s corresponde con un parámetro de inclinación en el perfil del núcleo.

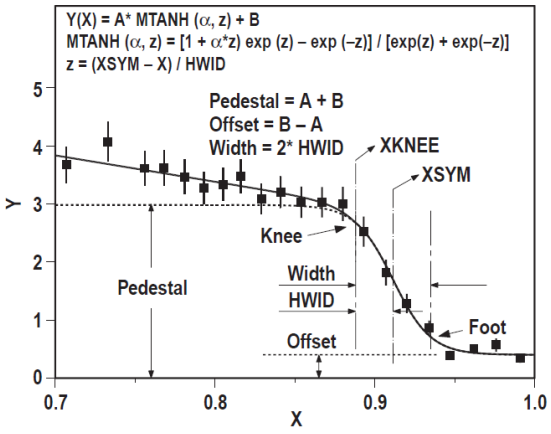


*Figura 2.7: Definición de la función tangente hiperbólica modificada. En la imagen, esta función se ajusta a los pares **(X;Y)**, donde **Y** es una cantidad, normalmente con una barrera de transporte, función de la coordenada espacial **X**. Los parámetros de ajuste de la función son la localización del centro de la barrera, **XSYM**; la mitad de la anchura de la barrera **HWID**; la altura en lo alto de la barrera (altura del pedestal), **Pedestal**; el desplazamiento de la barrera, **Offset**; y un parámetro que permite una transición suave al ajuste lineal del perfil del núcleo. La función explícita se encuentra definida en lo alto de la imagen. Esta función posee primeras derivadas continuas en todos sus puntos. Fuente: [21].*

De este modo, el ajuste de pedestal mtanh nos proporciona para cada perfil dos datos que son imprescindibles en este proyecto: la anchura y la altura del pedestal. La anchura Δ representa la distancia en términos de la coordenada espacial entre la zona alta del perfil y la zona baja, como vemos en la figura 2.7 se calcula como $\Delta = 2 \cdot \mathbf{HWID}$. La coordenada X del perfil, y por tanto la unidad de medida de la anchura del pedestal suele ser el flujo poloidal normalizado ψ_N, definido en 2.2. Esto se debe a que las cantidades que se miden en el eje Y de estos perfiles suelen ser constantes para un determinado valor de ψ_N, lo que provoca que este valor pueda actuar como una especie de «radio» para cantidades como densidad electrónica o temperatura, dada la naturaleza toroidal del tokamak [23]. Por otro lado, la altura del pedestal corresponde, con el valor **Pedestal** en la figura 2.7, es decir el punto más alto del perfil en el que se produce el cambio brusco de gradiente. En este caso, dependiendo de a qué perfil corresponda la representación la altura del pedestal tendrá una unidad de medida u otra, pero como acabamos de comentar, la densidad electrónica n_e y la temperatura T_e suelen ser los perfiles que mejor ilustran este pedestal, y por tanto dos variables imprescindibles en esta investigación.

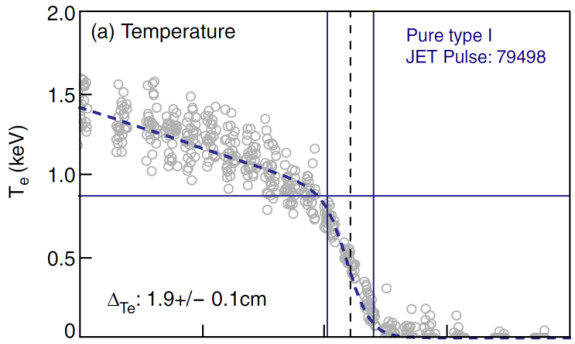


Figura 2.8: Medidas experimentales (circunferencias grises) de un pedestal de temperatura junto con su ajuste por la función mtanh (línea discontinua azul). Se presentan las medida entre $R = 3.7$ m y 3.9 m. La anchura del pedestal se marca entre dos líneas continuas verticales, y la altura mediante una línea continua horizontal. La línea discontinua vertical indica la posición del pedestal. Fuente: [24].

La función mtanh y su predecesora, llamada *tanhfit* por conveniencia, fueron desarrolladas para el análisis de la barrera de transporte del modo H. La función tanhfit se ajusta muy bien a los perfiles típicos del modo H de los bordes y los parámetros de ajuste se utilizan para cuantificar la altura la anchura y el gradiente máximo de las barreras de transporte en los perfiles de densidad, temperatura o presión de los bordes. Para estudiar las condiciones del modo L antes de la transición, es muy conveniente analizar los perfiles de borde en la región donde se forma la barrera de transporte del modo H. La función *tanhfit* también proporciona una manera conveniente y sistemática de hacerlo.

En la figura 2.8 podemos observar un ejemplo de medidas experimentales en el tokamak JET que forman un pedestal de temperatura y se ajustan mediante mtanh, observando claramente su altura y anchura. Los datos de esta figura también dan un ejemplo de las incertidumbres involucradas en las mediciones de los diferentes parámetros en plasmas de fusión. Medir la temperatura, la densidad y la presión en posiciones específicas del plasma no es una tarea sencilla y la dispersión en los datos puede ser muy grande.

2.2 Modelos de Pedestal de Plasmas

Dada la importancia del pedestal y el *H-mode* para el correcto confinamiento, y por tanto rendimiento, del plasma en un tokamak, elaborar modelos que puedan predecir ciertas cantidades tales como la anchura o la altura del pedestal antes de que el experimento ocurra puede ser crucial. De esta manera se han construido diferentes modelos para cumplimentar esta tarea, sobre todo de cara al proyecto ITER, donde la predicción del comportamiento del plasma podría permitirnos obtener energía de fusión neta.

2.2.1 EPED

El modelo EPED fue desarrollado en 2009 utilizando datos del tokamak DIII-D localizado en San Diego, Estados Unidos. Se trata de un modelo que predice la altura y anchura del pedestal antes de que el experimento se produzca [25].

Por lo general, la altura y anchura del pedestal del plasma crecen de forma conjunta hasta que se produce un ELM. Entonces, el modelo de estabilidad *peeling-ballooning* nos aporta una restricción tanto en altura como en anchura, o más específicamente, una relación entre la altura y la anchura en el valor máximo de la altura del pedestal. Sin embargo, tan solo esta restricción no es suficiente por si sola para determinar las dos incógnitas, anchura y altura del pedestal. Para proporcionar una segunda relación entre las incógnitas mencionadas, utilizaremos un argumento basado en el inicio de modos *ballooning* cinéticos y electromagnéticos (KBM) de turbulencia cercanos a un cierto valor crítico del gradiente de presión normalizada. Lo que quiere decir que esperamos que la anchura del pedestal tenga una fuerte dependencia de $\beta_{p,ped}$, es decir, de la beta poloidal del plasma en lo alto del pedestal. Esta $\beta_{p,ped}$ se calcula mediante la ecuación 2.6, tomando la densidad electrónica n_e y la temperatura T_e como sus respectivos valores en lo alto del pedestal del plasma, $n_{e,ped}$ y $T_{e,ped}$ respectivamente. Además, esta relación entre Δ y $\beta_{p,ped}$ también conlleva que la anchura del pedestal tenga una escasa dependencia de otros parámetros tales como la colisionalidad.

La hipótesis de que los ELM son provocados por los modos MHD ha existido desde el descubrimiento del *H-mode*. La importancia del acoplamiento *peeling-ballooning* ha sido notorio en los estudios con n (número modal toroidal) alto [26]. La extensión de esta teoría a un orden mayor y la implementación numérica en el codigo ELITE, permitieron un tratamiento cuantitativo de los modos de n intermedia con exitosas comparaciones respecto a experimentos [27; 28]. Paralelamente, un número de eficientes códigos de estabilidad MHD han sido desarrollados para estudios de estabilidad de los bordes del plasma con n intermedia [29; 30]. Estas publicaciones respaldan la implantación de la restricción del pedestal *peeling-ballooning* en el modelo.

De esta forma, el límite PB ofrece una restricción en la altura máxima del pedestal, además, sin entrar en detalle, los esquemas para el control o mitigación de modos ELM pueden ser optimizados para producir valores del pedestal cercanos a este límite pero sin excederlo. Entonces, este límite PB supone un factor de calidad para el pedestal en todos los regímenes, hacia el cual un esquema óptimo de mitigación o control de ELM tratará de aproximarse lo máximo posible. Es decir, nuestro plasma se confinara mejor y sus características serán más deseables cuanto más cerca se encuentre del límite PB (sin llegar a rebasarlo).

La estudios arriba citados discuten de manera extensa el modelo *peeling-ballooning*, pero basandose en reconstrucciones del equilibrio de descargas reales, por lo que solo han podido llevarse a cabo tras un cierto experimento. Sin embargo, para desarrollar un modelo predictivo, la estabilidad *peeling-ballooning* debería ser tan precisa como factible sin hacer uso de información solo disponible tras un experimento. Para este fin, el modelo utiliza los cálculos de estabilidad PB aumentando la altura del pedestal hasta encontrar el límite de estabilidad.

Tal y como se ha comentado anteriormente, la segunda relación que se utiliza para construir el modelo de predicción EPED está basada en la teoría de los modos *ballooning* cinéticos (KBM). Esta teoría propone un límite en el gradiente de presión normalizada α, que es proporcional a la beta poloidal en lo alto del pedestal $\beta_{p,ped}$ dividida por la

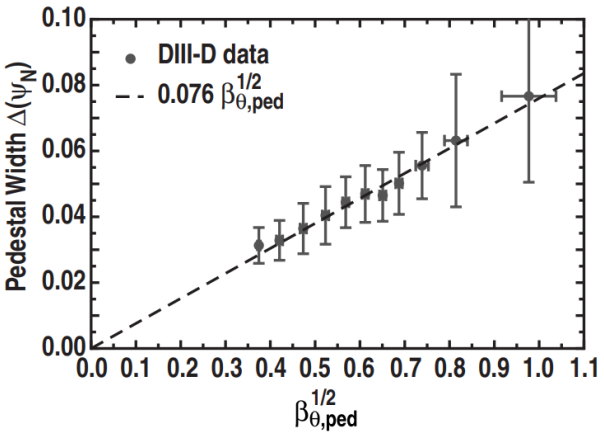


Figura 2.9: Medida del ancho del pedestal (Δ), correspondiente al tokamak DIII-D, como función de la raíz cuadrada de la beta poloidal en el pedestal. Encontramos una relación lineal con coeficiente 0.076 ilustrado por una línea discontinua. Fuente: [25].

anchura del pedestal. Por lo cual, suponiendo este límite impuesto por la tería KBM α_c constante, y que en esta zona la mayoría de densidad de corriente j proviene de la corriente *bootstrap*, siguiendo el desarrollo descrito en [25], llegamos a la ecuación 2.12, donde Δ representa la anchura del pedestal, c_1 un coeficiente que primeramente fue considerado constante (EPED1) y $\beta_{p,ped}$ la beta poloidal en lo alto del pedestal.

$$\Delta = c_1 \beta_{p,ped}^{\frac{1}{2}} \tag{2.12}$$

Esta ecuación representa la restricción que completa el modelo EPED y mediante la cual conseguiremos obtener ambas incógnitas: anchura y altura del pedestal del plasma. Además, esta correlación entre la anchura del pedestal y $\beta_{p,ped}$ que fue descubierta en el tokamak DIII-D [31; 32], también ha sido hallada en diferentes tokamaks entre los que se encuentran Alcator C-Mod y ASDEX-Upgrade [33].

En la figura 2.9 podemos observar la correlación comentada y el valor del coeficiente de proporcionalidad obtenido para los datos de DIII-D [25] que corresponde con $c_1 = 0.076$. En la mayoría de los tokamak este coeficiente es cercano a 0.1 pero no coincide en todos los tokamak ni tan siquiera en todos los experimentos.

En la figura 2.10 observamos como a partir de la intersección entre el límite de estabilidad del modelo *peeling-ballooning* y la restricción encontrada para el modelo KBM

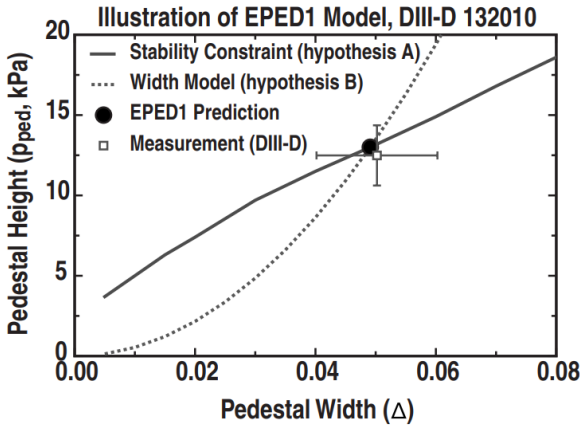


Figura 2.10: Ilustración del mecanismo del modelo primigenio EPED1. La línea continua, calculada mediante el código ELITE es la restricción de estabilidad del modelo peeling-ballooning. La línea discontinua corresponde con la relación de la anchura obtenida del modelo KBM: $\Delta = 0.076\beta_{p,ped}^{\frac{1}{2}}$*. La intersección de ambas, que se muestra con un circulo negro, nos ofrece la predicción de EPED1 de la altura y anchura del pedestal. Un experimento llevado a cabo con el mismo conjunto de datos de entrada produce el cuadrado, con su pertinente precisión. Fuente: [25].*

obtenemos nuestra predicción para la altura y anchura del pedestal, coincidiendo en este caso con los datos experimentales dentro de sus márgenes de error.

De esta forma concluimos que el modelo EPED tiene como datos de entrada los siguientes 8 parámetros: campo magnético toroidal B_T, intensidad de corriente I_p, radio mayor R_0, radio menor a, elongación κ, triangularidad δ, beta global β y densidad electrónica en lo alto del pedestal $n_{e,ped}$. Y como datos de salida: la altura del pedestal, representado por la presión p_{ped} o equivalentemente por la temperatura T_{ped}, y la anchura del pedestal Δ. De esta forma este modelo predice las cantidades deseadas.

2.2.2 EUROPED

Tal y como se ha comentado en la sección 2.1.2 en el plasma de un tokamak donde rige el *H-mode* existen dos regiones diferenciadas: el núcleo y el pedestal. El perfil de presión total resultante (y la potencia de fusión dado que depende de la forma $P_{fus} \propto p^2$) es una combinación de estas dos regiones. Sin embargo estas regiones no pueden predecirse de manera independiente, debido a que el pedestal afecta al núcleo y el núcleo afecta al pedestal. Dado que el perfil de temperatura del núcleo está generalmente dirigido por tur-

bulencias cuyo impulso es proporcional al gradiente de temperatura normalizado $\Delta T/T$, y que la altura del pedestal fija la condición de contorno, entonces la temperatura máxima alcanzable en el núcleo depende fuertemente del valor de pedestal. Además, aumentar la presión del núcleo aumenta el fenómeno *Shafranov-shift*, que consiste en el desplazamiento del centro de las superficies de flujo, lo cual tiene un efecto estabilizador en el límite impuesto por el modelo *peeling- ballooning* [34]. Este bucle que se retro-alimenta hace que sea esencial resolver el pedestal y el núcleo de manera auto-consistente.

EuroPED es un modelo de pedestal desarrollado a partir de EPED, donde como sabemos, el pedestal en el perfil de presión del plasma está limitado por la combinación de los modelos *peeling-ballooning* y KBM. En EuroPED se utiliza la relación que proviene de EPED1 [25] entre la anchura del pedestal y el parámetro beta poloidal en lo alto del pedestal: $\Delta = 0.076\sqrt{\beta_{p,ped}}$. También asumimos que la anchura del pedestal Δ es idéntica tanto en perfiles de temperatura como de densidad en unidades de flujo poloidal normalizado. Esta relación se ha demostrado que funciona correctamente para pedestales JET-ILW (tokamak JET con paredes interiores que replican las del experimento en construcción ITER) cuando se inyectan niveles bajos de combustible, sin embargo cuando se inyectan niveles superiores esta relación subestima el valor Δ [35]. De esta manera los casos donde se inyectan cantidades altas de combustible no resultan adecuados para modelos basados en la estabilidad *peeling-ballooning* (tal como EuroPED), porque suelen ser estables en los modos PB. Teniendo esto en cuenta el código EuroPED toma un rango de pedestales donde varían las anchuras con sus correspondientes alturas (según la citada relación), entonces la estabilidad del equilibrio se resuelve con un código MHD de estabilidad llamado HELENA.

En ese punto el equilibrio calculado se pasa por un código de estabilidad PB que puede ser ELITE [14] o MISHKA [15]. La elección debería ser arbitraria pues ambos códigos han sido contrastados y comparados y su funcionamiento es óptimo para números modales toroidales $n > 4$ [27]. Este código se encarga de calcular las tasas de crecimiento γ de una lista de modos toroidales de perturbación, también introducido como parámetro para cada equilibrio. En ese momento se aplica una condición crítica a las tasas de crecimiento para determinar cuando un modo se ha vuelto inestable. El valor crítico γ_{crit} se muestra en

la ecuación 2.13 definido como en [36], donde ω^* representa la frecuencia diamagnética media y n el número modal toroidal.

$$\gamma_{crit} = \begin{cases} \omega^*/2 & n \leq 10 \\ \omega^*/2(n=10) & n > 10 \end{cases} \tag{2.13}$$

La predicción del pedestal corresponde con el pedestal en el que $\gamma = \gamma_{crit}$, es decir, el

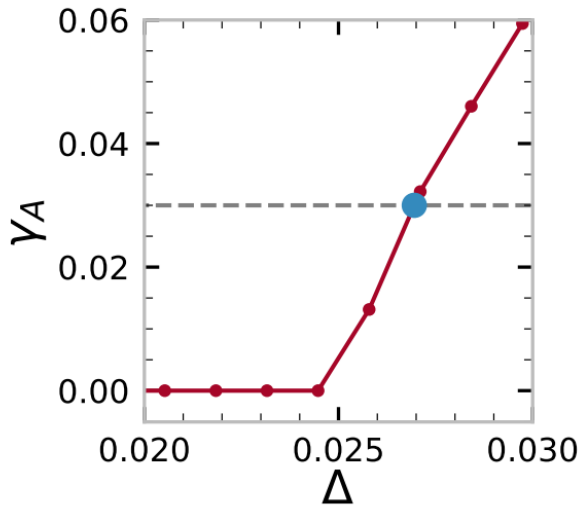


Figura 2.11: Ejemplo de tasas de crecimiento del modo más inestable frente a la anchura de pedestal proveniente de EuroPED. La linea discontinua gris representa la condición crítica y el punto azul el pedestal predicho. En la figura γ_A se define como $\gamma_A = \gamma/\omega_A$ donde ω_A corresponde con la frecuencia de las ondas de Alfven en el núcleo del plasma. Fuente: [17]

modo PB que más rápido crece sin ser considerado inestable. O lo que es lo mismo, el pedestal predicho corresponde con la anchura de pedestal más baja en la que un modo es inestable, la cual tiene una altura de pedestal que relacionamos con el criterio KBM, así obtenemos la predicción completa de anchura y altura del pedestal. Podemos observar en la figura 2.11 un ejemplo de esto, los puntos rojos representan las tasas de crecimiento calculadas para los equilibrios correspondientes a cada anchura del pedestal. Podemos ver como a partir de un cierto límite, los equilibrios son inestables dada la elevada tasa de crecimiento, por lo que la predicción se encuentra en el propio límite.

El modelo EPED, por otro lado, posee dos parámetros que no son generalmente conocidos antes del experimento: beta plasmática total β y la densidad del pedestal $n_{e,ped}$,

a menos que se oriente el experimento para que estos valores estén fijados. A fin de ganar cierta generalización EuroPED trata de solucionar este problema. Para el caso de la densidad de pedestal se proponen dos alternativas a la hora de predecir la densidad del pedestal a priori. La primera está basada en la parametrización de la gigantesca base de datos de experimentos en JET-ILW [37]. En esta parametrización se usan tan solo parámetros de ingeniería que son conocidos antes del experimento y son útiles para predecir pedestales en JET-ILW. La expresión que se obtiene para calcular $n_{e,ped}$ la podemos observar en la ecuación 2.14 (desarrollada en [38]) donde I_p es la corriente del plasma en MA, B_t es el campo magnético toroidal (T), δ es la triangularidad del plasma, P_{NBI} es la potencia de calentamiento en MW y ϕ_e es la tasa de inyección combustible ($10^{22}s^{-1}$). Se puede apreciar, por su coeficiente que la mayor influencia se encuentra en la corriente del plasma.

$$n_{e,ped}[10^{19}m^{-3}] = 11.4 \cdot I_p^{1.38} \cdot B_t^{-0.42} \cdot P_{NBI}^{-0.25} \cdot \delta^{0.71} \cdot \phi_e^{0.11} \tag{2.14}$$

Sin embargo, esta alternativa tiene utilidad limitada en otros dispositivos ya que no contiene ningún modelo físico. Al contrario que en la segunda alternativa, la cual está basada en principios físicos. Se trata del modelo de penetración neutra [39]. Este modelo asume que toda la inyección de combustible proviene del borde del plasma y que el coeficiente de difusión D es constante en el espacio, lo que conlleva una relación entre la anchura Δ_{ne} y altura $n_{e,ped}$. La podemos observar en la ecuación 2.15 donde V_n es la velocidad de las partículas neutras, σ_i es la sección eficaz del impacto del electrón en la ionización, V_e es la velocidad térmica del electrón en lo alto del pedestal y E es un factor de proporcionalidad.

$$\Delta_{ne} = 2V_n/(\sigma_i V_e E n_{e,ped}) \tag{2.15}$$

Tal como ocurría en EPED1, en EuroPED asumimos que $\Delta_{ne} = \Delta_{Te} = \Delta$. Así que siguiendo los razonamientos expuestos en [38] podemos obtener la altura del pedestal de densidad a través de la anchura y la temperatura del pedestal. El cálculo final se hace en un bucle iterativo dado que la presión total del pedestal para una determinada anchura del pedestal está restringida por la condición $\Delta = 0.076\sqrt{\beta_{p,ped}}$, que requiere ajustar la tem-

peratura del pedestal como respuesta a una recientemente calculada densidad del pedestal para mantener $\beta_{p,ped}$ fijada. También asumimos que la temperatura y la densidad del pedestal están alineadas, y aunque esta asunción no es cierta, este desajuste se contrarresta en JET-ILW por estar el pedestal de densidad más cerca de la *separatrix* que el pedestal de temperatura [40]. Por un lado, el efecto de desplazar el pedestal de densidad provoca que la posición de gradiente de presión máximo se desplace hacia la *separatrix*, lo que tiene un efecto desestabilizante en los modos PB. Por otro lado, el hecho de que los perfiles estén desalineados reduce el gradiente máximo de presión para una determinada altura del pedestal, aumentando de este modo la estabilidad. Estos dos efectos se contrarrestan casi completamente y provoca que los posibles desplazamientos del pedestal de densidad puedan ser ignorados en las predicciones. Finalmente tanto en EPED como EuroPED asumimos $T_i = T_e$ en el pedestal, siendo T_e la temperatura de los electrones y T_i la de los iones.

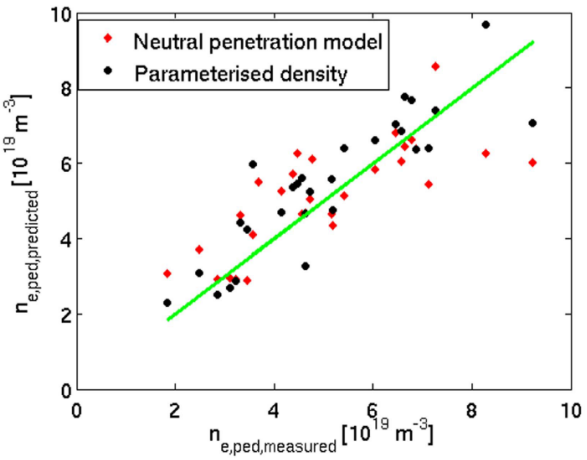


Figura 2.12: Predicción de densidad mediante el código EuroPED en datos de JET-ILW, usando los dos modelos: parametrización y penetración neutral. La diagonal indica los puntos donde la predicción es igual al experimento. Fuente: [17].

De este modo, tal y como observamos en la figura 2.12 tanto el modelo de parametrización como el de penetración neutra predicen aceptablemente los valores de la densidad en el pedestal $n_{e,ped}$.

Como he mencionado anteriormente, en el modelo EPED la β global se asume como conocida antes del experimento. Mientras que esto podría ser una aproximación razonable

para experimentos donde se ajusta la potencia para alcanzar un valor determinado de β, a menudo en experimentos la forma de la onda de potencia de calentamiento total se establece de antemano y la β resultante es un resultado del experimento. Además, cuando se trata de alcanzar los máximos rendimientos, los experimentos se llevan a cabo a la máxima potencia posible, no ofreciendo flexibilidad a la hora de ajustarla.

En los casos donde se conoce la potencia de calentamiento en vez de β, los perfiles de temperatura del núcleo T_i, T_e, han de predecirse usando un modelo de transporte en estado estacionario para iones y electrones. Este se encuentra desarrollado en [41], y se corresponde con la ecuación 2.16. En él toman parte la difusividad χ, la derivada radial del volumen plasmático $V' = dV/d\rho$, densidad n, flujo de calor q_e y coordenada radial ρ.

$$\frac{\partial T_{e,i}}{\partial \rho} = -\frac{q_{e,i}}{V' \|\nabla \rho\|^2 n_{e,i} \chi_{e,i}} \tag{2.16}$$

Para obtener la difusividad χ se utiliza el modelo de transporte *Bohm-gyroBohm*[1] [42] que ha demostrado ser adecuado para el núcleo de JET-ILW [43].

Este método es implementado en EuroPED especificando la amplitud de la fuente de calor y su perfil y resolviendo la ecuación 2.16 con los pedestales de temperatura como condición de contorno. Después esto se usa para calcular el perfil de densidad del núcleo. El proceso se repite hasta que converge. Un nuevo equilibrio se calcula con los nuevos perfiles, asumiendo que la corriente es una combinación de corriente impulsada por inducción completamente difundida calculado a partir del perfil de conductividad neoclásico y la corriente *bootstrap* calculada usando fórmulas en [44; 45]. Esto se continúa hasta que los perfiles y el equilibrio son auto-consistentes. Este paso se realiza para un rango de hipotéticos pedestales (con valores de Δ y $\beta_{p,ped}$ consistentes). La estabilidad de los equilibrios se resuelve y, tal y como se indica anteriormente, la predicción final se obtiene en le punto crítico donde $\gamma = \gamma_{crit}$. En la figura 2.13 podemos ver un esquema del funcionamiento de como la predicción completamente auto-consistente núcleo-pedestal funciona en el código EuroPED completamente autónomo.

[1] Aunque en la actualidad existen modelos mucho más precisos que tienen en cuenta el transporte turbulento dentro del núcleo, este modelo proporciona una buena estimación de orden cero a efectos de estimar la β global. Dicho esto, es probable que haya discrepancias entre la β predicha por este método y la β experimental, como se observará en la base de datos utilizada en este proyecto.

Dado que todos los cálculos se realizan en el interior de la iteración de EuroPED, es-

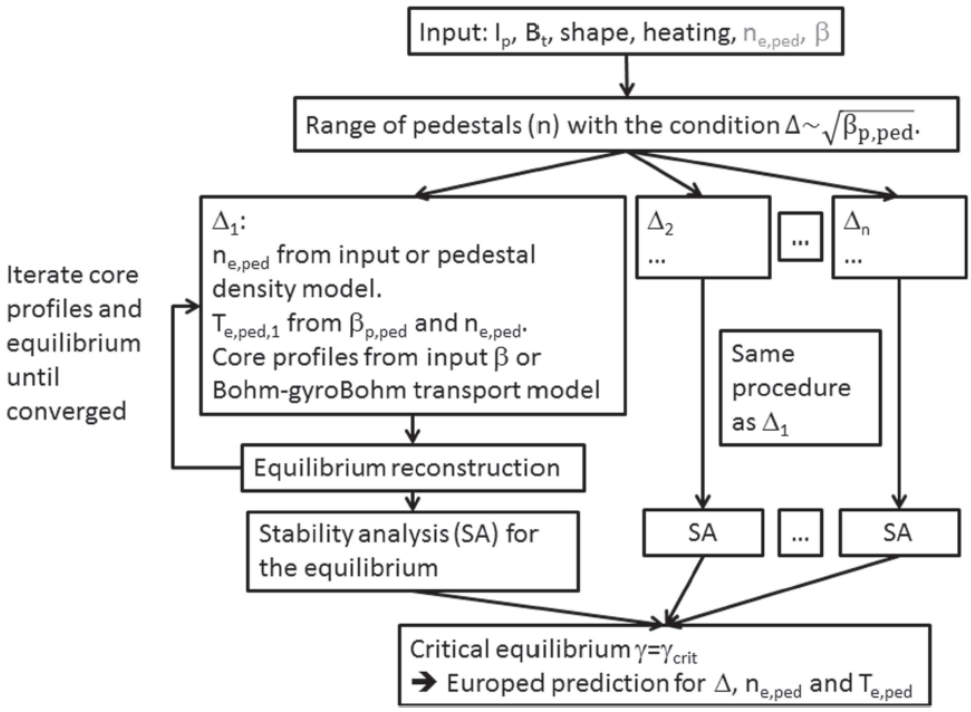


Figura 2.13: Representación del flujo de trabajo de EuroPED. Las lineas paralelas corresponden con cada pedestal dado un determinado Δ*, las iteraciones se realizan en paralelo para cada pedestal independientemente. La predicción final corresponde con el pedestal que es estable PB marginalmente* ($\gamma = \gamma_{crit}$)*. Fuente: [38].*

to resulta un método relativamente rápido, además cada pedestal hipotético se resuelve paralelamente para aumentar el rendimiento. Mientras que esto puede ofrecernos una estimación aproximada del pedestal y la β total, si combinamos la predicción del pedestal con un código de transporte apropiado que resuelva el perfil de calor y el intercambio de energía entre partículas, podemos obtener resultados mas sofisticados.

En definitiva, el modelo EuroPED toma las siguientes 8 variables de entrada: intensidad de corriente I_p, campo magnético toroidal B_t, triangularidad δ, elongación κ, radio mayor R_0, radio menor a, potencia total P_{tot}, tasa de inyección de combustible *Gas* y numero atómico efectivo Z_{eff}. Sus variables de salida son la anchura del pedestal Δ, la altura del pedestal para la temperatura Te_{ped} y la densidad electrónica en el pedestal ne_{ped}, sin olvidar que durante el proceso se calcula de manera auto-consistente el parámetro plasmático β. Este sería un resumen final de lo que el modelo EuroPED necesita y de las

predicciones que ofrece.

De todo lo visto hasta ahora se puede concluir que las ventajas de EuroPED sobre EPED son numerosas. Por un lado, EPED es dependiente de variables que solo se conocen tras haber realizado el experimento, estas son la densidad electrónica en el pedestal ne_{ped} y la β plasmática total. Y por lo tanto solo es válido en experimentos que tienen como objetivo ciertos valores para estas variables, lo que supone un gran obstáculo en el desarrollo de los experimentos. EuroPED soluciona este problema calculando estas variables de manera auto-consistente y ofreciendo resultados fiables. Por otro lado, EPED fue pensado tan solo para el dispositivo DIII-D, mientras que EuroPED se gestó en torno al tokamak JET y se ha testado en diferentes dispositivos con éxito. Además EPED está incorporado en el seno de EuroPED, por lo que se podría considerar una extensión del mismo.

2.2.3 Modelo Sustituto

A pesar de toda su funcionalidad, tanto el modelo EuroPED como EPED tienen un gran inconveniente, el coste computacional y su velocidad de computación. Por ser modelos tan complejos y que requieren de una inmensidad de cálculos, estos modelos son lentos y por lo tanto dejan de ser adecuados en muchas investigaciones, por ejemplo en experimentos en tiempo real.

Por suerte, el objetivo principal de esta investigación es acelerar los modelos EPED y EuroPED, y esto se llevará a cabo mediante un modelo sustituto basado en técnicas de *machine learning*. Pues bien, los modelos sustitutos sirven para capturar los comportamientos más cruciales de los modelos y poder aplicarlos a *posteriori* a experimentos donde no hubiera sido posible de otra manera por su gran carga computacional.

Los modelos sustitutos o *surrogate models* en inglés, son modelos que consiguen acelerar y tomar las dependencias y comportamientos más cruciales del modelo original, el

cual suele ser demasiado pesado computacionalmente y complicado. A su vez, también consiguen mostrar una perspectiva más simplista que a menudo muestra comportamientos ocultos o la física subyacente tras el problema, lo cual suele ser bastante útil.

En nuestro caso, los modelos sustitutos de EPED y EuroPED servirán para aplicar estos modelos originales a diferentes problemas con mayor agilidad, ya que en muchos casos resultan demasiado lentos y complejos.

Además de esto, otra gran traba en nuestros modelos de pedestal del plasma, es la obtención de datos. Debido a que un tokamak es un dispositivo muy complejo y de grandes dimensiones, ponerlo en funcionamiento no es una tarea sencilla y consume grandes cantidades de energía. Por ello, la obtención de datos se vuelve costosa. Mediante los modelos sustituos podemos generar datos provenientes de EPED y EuroPED a mayor velocidad, lo cual es muy útil pues uno de los cuellos de botella más comunes suele ser la falta de datos.

De esta manera, los modelos sustitutos, especialmente los basados en inteligencia artificial y ciencia de datos (los desarrollados en este trabajo), se presentan como una alternativa de gran valor añadido, pues además de agilizar los modelos originales, se obtiene nueva información sobre los mismos y la obtención de datos se simplifica.

2.3 *Machine Learning* y Redes Neuronales

En las últimas décadas, el campo de la inteligencia artificial ha experimentado un gran auge, tanto en la industria como en la investigación. Las técnicas englobadas por esta disciplina han resultado ser muy útiles en campos muy diversos: la medicina [46], la política, innumerables sectores tecnológicos, la investigación científica, etc. Todos se han hecho eco del *boom* de la inteligencia artificial. Los beneficios son diversos: detección de fraudes, predicciones en distintos ámbitos, optimización de procesos, reconocimiento de voz, visión artificial [47] y un largo etcétera. Sin embargo, su labor se puede resumir en interpretar información que se le escapa al ojo humano o a los mecanismos tradicionales, y ponerla al servicio del ser humano, o simplemente liberarle de una tarea que un dispositivo puede hacer de manera automática.

Dentro de la inteligencia artificial, nos interesa el campo del *Machine Learning* o aprendizaje automático. Este trata el estudio de los algoritmos informáticos que pueden mejorar automáticamente a través de la experiencia y el uso de datos. Los algoritmos de aprendizaje automático construyen un modelo basado en una muestra de datos, conocidos como datos de entrenamiento, para hacer predicciones o tomar decisiones sin estar programados explícitamente para hacerlo.

A través de algoritmos, esta disciplina dota a los ordenadores de la capacidad de identificar patrones en datos masivos y elaborar predicciones (análisis predictivo). Este aprendizaje permite a los ordenadores realizar tareas específicas de forma autónoma, es decir, sin necesidad de ser programados. El término se utilizó por primera vez en 1959. Sin embargo, ha ganado relevancia en los últimos años debido al aumento de la capacidad de computación y al *boom* de los datos [48].

De este modo, los algoritmos de *Machine Learning* se dividen en tres categorías:

- **Aprendizaje supervisado**: se le muestran al ordenador datos de entrada de ejemplo y la salida que deseamos, el objetivo es la generalización de este aprendizaje a datos diferentes a los de entrenamiento.
- **Aprendizaje no supervisado**: Los datos que se le ofrecen al algoritmo no tienen etiquetas de las que pueda aprender, por lo que ha de aprender y captar la estructura de los datos por sí mismo. Muy útil a la hora de encontrar patrones en los datos ocultos al ojo humano.
- **Aprendizaje por refuerzo**: El algoritmo interactúa con un entorno dinámico en donde ha de realizar una determinada tarea, recompensándose su consecución. Un ejemplo podrían ser los coches auto-dirigidos.

El *machine learning* engloba una gran variedad de métodos: árboles de decisión, máquinas de vectores de soporte, algoritmos de agrupamiento, redes neuronales artificiales, etc. Sin embargo, este proyecto se centra de manera muy esencial en uno de ellos, las redes neuronales.

2.3.1 Redes Neuronales

Las redes neuronales artificiales, referidas normalmente como «redes neuronales», han sido desde sus inicios motivadas por el hecho de que el cerebro humano computa de una manera totalmente diferente a los ordenadores convencionales. El cerebro, visto como un sistema de procesamiento de la información es, en efecto, un ordenador altamente complejo, no lineal y que trabaja en paralelo. Además tiene la capacidad de organizar sus estructuras constituyentes, conocidas como "neuronas", para llevar a cabo ciertos procesos (como por ejemplo reconocimiento de patrones, percepción y control motor) de una manera mucho más rápida que los ordenadores más potentes hoy en día.

Un ejemplo de esto es el sonar de un murciélago, se trata de un sistema activo de localización sonora. Además de proporcionar información sobre la distancia al objetivo, el sonar del murciélago recopila información sobre la velocidad relativa del objetivo, el tamaño del mismo y de sus componentes, y el azimut y la elevación del objetivo [49]. Los complejos procesos neuronales necesarios para obtener esta información del objetivo ocurren dentro de un cerebro del tamaño de una ciruela. De hecho, un murciélago gracias a su sonar es capaz de perseguir y capturar a su presa con una facilidad y una tasa de éxito que serían la envidia de cualquier radar o sonar construido por el ser humano.

Con esta introducción, podemos definir a las redes neuronales tal y como se presentan en [50].
Una red neuronal es un procesador masivo paralelamente distribuido constituido por unidades de procesamiento simples, el cual tiene una propensión natural de almacenar conocimiento en base a la experiencia y ponerlo a disposición para su uso. Se asemeja al cerebro en dos aspectos:

1. El conocimiento es adquirido por la red de su entorno a través de un proceso de aprendizaje.

2. Las fuertes conexiones interneuronales, conocidas como pesos sinápticos, se utilizan para almacenar el conocimiento adquirido.

De esta manera, para alcanzar un buen rendimiento, las redes neuronales emplean la interconexión masiva de simples células de computación llamadas «neuronas» o «unidades de procesamiento». El procedimiento utilizado para llevar a cabo el proceso de aprendizaje se denomina algoritmo de aprendizaje, cuya función es modificar los pesos sinápticos de la red de una manera ordenada para alcanzar un objetivo deseado.

Beneficios de las redes neuronales

Es apreciable que la red neuronal obtiene su poder computacional a través de, en primer lugar, su masiva estructura paralelamente distribuida y, en segundo lugar, su habilidad para aprender y por tanto generalizar. En este caso, la generalización hace referencia a la red neuronal produciendo salidas razonables para valores de entrada con los que no ha sido entrenada (proceso de aprendizaje). Estas dos capacidades relacionadas con el procesamiento de la información hacen posible que las redes neuronales puedan resolver complejos problemas que de otra forma serían irresolubles.

De esta manera, el uso de redes neuronales ofrece las siguientes ventajas:

- **No linealidad**: Las neuronas pueden ser lineales o no lineales. Esta propiedad es muy importante especialmente si el proceso físico subyacente a la generación de los datos de entrada es no lineal.
- **Relación entrada-salida**: Este comportamiento es de especial importancia en el aprendizaje supervisado, donde el cambio en los pesos de la red se lleva a cabo en función de datos de entrenamiento etiquetados según la salida. Es decir, el conjunto de datos de entrenamiento consiste en ciertos valores de entrada y su salida deseada. La red ajusta entonces sus pesos para minimizar la diferencia entre la salida deseada y la salida de la propia red.
- **Adaptabilidad**: Se evidencia en su capacidad para adaptar los pesos sinápticos a los cambios de su alrededor. Particularmente, una red neuronal entrenada para operar en un entorno específico puede ser fácilmente re-entrenada para lidiar con pequeños cambios en las condiciones operacionales del entorno.

Modelos de neurona

La neurona es una unidad de procesamiento de la información fundamental para el funcionamiento de la red neuronal. El diagrama de la figura 2.14 ilustra el modelo de una neurona, que constituye la base del diseño de las redes neuronales artificiales. Aparte de

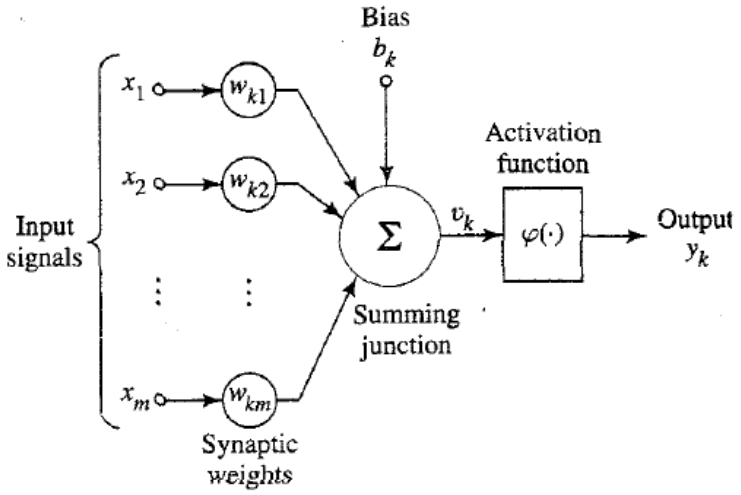


Figura 2.14: Modelo no lineal básico de la neurona. Fuente: [50].

los datos de entrada x_j (*input*) y salida y (*output*), se pueden identificar tres elementos básicos:

1. Un conjunto de **sinapsis** o hilos conectores, cada uno de ellos caracterizado por un **peso**. Particularmente, una señal x_j en la entrada de la sinapsis j conectada a la neurona k se multiplica por el peso sináptico w_{kj}.

2. **Agregador** para sumar los valores de entrada ponderados con sus pesos determinados por las sinapsis de la neurona. La operación que se describe es una combinación lineal.

3. **Función de activación**: Se trata de una transformación de la agregación que limita la amplitud de la salida de la neurona, por lo general no es lineal. Siendo $\varphi : \mathcal{I} \longrightarrow \mathcal{O}$ una función bien definida entre el conjunto de valores de la agregación $\mathcal{I}$ y el conjunto de salida $\mathcal{O}$

El modelo neuronal de la figura 2.14 incluye el sesgo o *bias* denotado como b_k para la neurona k. El sesgo b_k tiene el efecto de aumentar o disminuir el *input* de la función de

activación. De esta manera, describimos a la neurona k mediante las ecuaciones 2.17 y 2.18, donde x_1, x_2, ..., x_m son los valores de entrada; w_{k1}, w_{k2}, ..., w_{km} son los pesos sinápticos de la neurona k; u_k es la salida del agregador lineal; b_k es el sesgo; $\varphi(\cdot)$ es la función de activación; e y_k es la salida de la neurona k.

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{2.17}$$

$$y_k = \varphi(u_k + b_k) \tag{2.18}$$

El uso del sesgo b_k tiene como efecto una transformación de la salida del agregador lineal de la forma $v_k = u_k + b_k$.

Sin embargo, si ponemos nuestra atención tan solo en una neurona, podemos describir la salida del agregador lineal junto al sesgo como en la ecuación 2.19, donde denotamos el sesgo como b. Este se interpreta como un estímulo interno en la neurona, de manera que la agregación es una transformación afín de las variables de entrada. Dicho esto exploraremos algunas propiedades de esta agregación.

$$z = \left(\sum_{j=1}^{m} w_{ixi} \right) + b \tag{2.19}$$

Denotaremos como $(\mathbf{a}, \mathbf{b})$ a la concatenación de los vectores $\mathbf{a} = (a_1, \ldots, a_N) \in \mathbb{R}^N$ y $\mathbf{b} = (b_1, \ldots, b_M) \in \mathbb{R}^M$, por lo que obtenemos el resultado de la ecuación 2.20.

$$(\mathbf{a}, \mathbf{b}) = (a_1, \ldots, a_N, b_1, \ldots, b_M) \in \mathbb{R}^{N+M} \tag{2.20}$$

Utilizamos esta notación para representar la ecuación 2.19 vectorialmente, sin el sesgo representada como $z = \mathbf{w} \cdot \mathbf{x}$ y añadiendo el sesgo $z = \mathbf{w} \cdot \mathbf{x} + b$, o equivalentemente $z = (\mathbf{w}, b) \cdot (\mathbf{x}, 1)$, siendo $\cdot$ el producto escalar usual en $\mathbb{R}^m$

Si definimos $z = z(\mathbf{w}, \mathbf{x}, b)$ como una función $z : \mathbb{R}^{2m+1} \longrightarrow \mathbb{R}$ de $\mathbf{w}$, $\mathbf{x}$ y b, podemos demostrar los siguientes resultados [51].

Proposición 2.1 *La función de agregación z es continua y suprayectiva en la topología*

usual de $\mathbb{R}^{2m+1}$.

Demostración: La continuidad es en este caso evidente ya que podemos construir z como resultado de la composición de las funciones continuas producto $f(x,y) = xy$ y suma $f(x,y) = x+y$. De otra manera, estas funciones son suprayectivas lo cual implica que su composición también lo será.

Proposición 2.2 *Las neuronas con función de agregación* $z = z(\mathbf{w},\mathbf{x},b)$ *y pesos fijados* $\mathbf{w} = \mathbf{w_0}$ *y sesgo fijado* $b = b_0$ *verifican que z es lipschitziana.*

Demostración: Dadas dos entradas de la neurona $\mathbf{x_1}$ y $\mathbf{x_2}$ cualesquiera con agregaciones $z_1 = (\mathbf{w},b)\cdot(\mathbf{x_1},1)$ y $z_2 = (\mathbf{w_0},b_0)\cdot(\mathbf{x_2},1)$. Se tiene:

$$|z_1 - z_2| = |(\mathbf{w_0},b_0)\cdot(\mathbf{x_1},1) - (\mathbf{w_0},b_0)\cdot(\mathbf{x_2},1)| = (\mathbf{w_0},b_0)\cdot(\mathbf{x_1}-\mathbf{x_2},0)$$
$$\leq \text{máx}\,(|w_{0,1}|,...,|w_{0,N}|,|b_0|)\cdot||x_1 - x_2||$$

Por lo que la función de agregación z es lipschitziana de constante

$$(\text{máx}\,(|w_{0,1}|,...,|w_{0,N}|,|b_0|)).$$

.

Proposición 2.3 *Las neuronas con función de agregación* $z = z(\mathbf{w},\mathbf{x},b)$ *y entrada fijada* $\mathbf{x} = \mathbf{x_0}$ *verifican que z es lipschitziana.*

Demostración: Análogo al caso anterior, solo que con $\mathbf{x_0}$ tomando el papel de $(\mathbf{w_0},b)$ en la demostración anterior y viceversa.

Podemos deducir entonces que fijar los pesos o las entradas de la neurona no afecta a la suprayectividad de la agregación.

A causa de estos resultados podemos enunciar dos propiedades de la agregación de las neuronas:

- La agregación de dos valores de entrada x_1 y x_2 similares por parte de una neurona con pesos fijados es también similar.

- Es posible variar el resultado de la agregación de una neurona a cada entrada de manera controlada hacia cualquier valor. Lo cual refleja la adaptabilidad de la misma.

Por último, hablaremos sobre **la función de activación**. Tras la agregación de la información de entrada y los pesos en la neurona, esta se modifica una última vez antes de propagarse a la siguiente capa o neurona de la red. Esta modificación se corresponde con la función de activación. Podemos expresar la salida de la neurona mediante la composición de las funciones de agregación z y activación φ como vemos en la ecuación 2.21.

$$y = (\varphi \circ z)(\mathbf{w}, \mathbf{x}, b) \tag{2.21}$$

Asimismo, las propiedades de la función agregación z son tan idóneas que provocan que las características de la salida de la neurona tan solo estén especificadas por la función de activación, mientras que las relaciones entre las variables de entrada tan solo dependen de la agregación. De esta forma, la suprayectividad de z provoca que la composición $(\varphi \circ z)$ sea suprayectiva en su imagen $Im\,\varphi$, sin embargo la continuidad de la salida de la neurona es determinada por la continuidad de φ.

Generalizamos este modelo de neurona a m entradas y a n salidas en lugar de una salida. Esto se realiza empleando cada una de las neuronas independientes que comparten los datos de entrada, pero no los pesos ni la activación. Esto da lugar a un mapeo general de $\mathbb{R}^m$ en $\mathbb{R}^n$ donde la componente k-ésima de la salida es la mostrada en la ecuación 2.22

$$y_k = f_k\Big(\sum_{j=1}^{m} w_{kj}x_j + b_k\Big) \tag{2.22}$$

Si usamos notación matricial obtenemos la ecuación 2.23, donde W es la matriz de pesos y $\mathbf{b}$ el vector de sesgos.

$$\mathbf{y} = \mathbf{f}(W\mathbf{x} + \mathbf{b}) \tag{2.23}$$

El principal papel de la función de activación es añadir una contribución no lineal al procesado de datos. Esto permite a la red neuronal resolver problemas de naturaleza no lineal,

ya que está función es la que le confiere la característica de aproximador universal de funciones que se detallará en el apartado 2.3.1.

Aunque, la función de activación puede presentar cualquier forma funcional, existen familias de funciones de activación que han demostrado resultados notables en su aplicación en redes neuronales. A continuación se describen las dos funciones de activación más relevantes y comúnmente utilizadas [52]:

- **Sigmoide**: La función sigmoide se podría caracterizar por ser la más antigua y popular, se define como en la ecuación 2.24, donde e denota el número de Euler.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.24}$$

  Una neurona en la que se utiliza la función sigmoide como función de activación se denomina neurona sigmoide. En la figura 2.15(b) podemos ver que $\sigma(z)$ actúa de manera que comprime nuestra salida en un rango de 0 (para valores negativos) a 1 (para valores positivos).
  Aunque, las funciones sigmoides fueron la base de la mayoría de las redes neuronales inicialmente, en los últimos años han perdido popularidad. La razón es que las redes neuronales de muchas capas se vuelven muy difíciles de entrenar dado el problema de desaparición de gradiente. En su lugar, la mayoría de las redes neuronales actuales utilizan otro tipo de función de activación denominada *rectified linear unit* o ReLU.

- **ReLU**: Se define simplemente como vemos en la ecuación 2.25, es decir, dejan inalterados los valores positivos y llevan a 0 los negativos. Podemos observar su forma en la parte izquierda de la figura 2.15(a).

$$ReLU(x) = max(0,x) \tag{2.25}$$

  Una variación de esta, denominada **LeakyReLU**, se trata de una modificación que la hace creciente, donde en los valores negativos tenemos en vez de 0 una recta con pendiente a como se observa en la parte derecha de la figura 2.15(a) y en su

definición en la ecuación 2.26. Ambas ReLU y LeakyReLU son las funciones de activación más utilizadas hoy en día, y en concreto LeakyReLU es la que utilizaremos en nuestra investigación.

$$LeakyReLU(x) = \begin{cases} ax, & \text{si } x \leq 0 \\ x, & \text{si } x > 0, \end{cases} \tag{2.26}$$

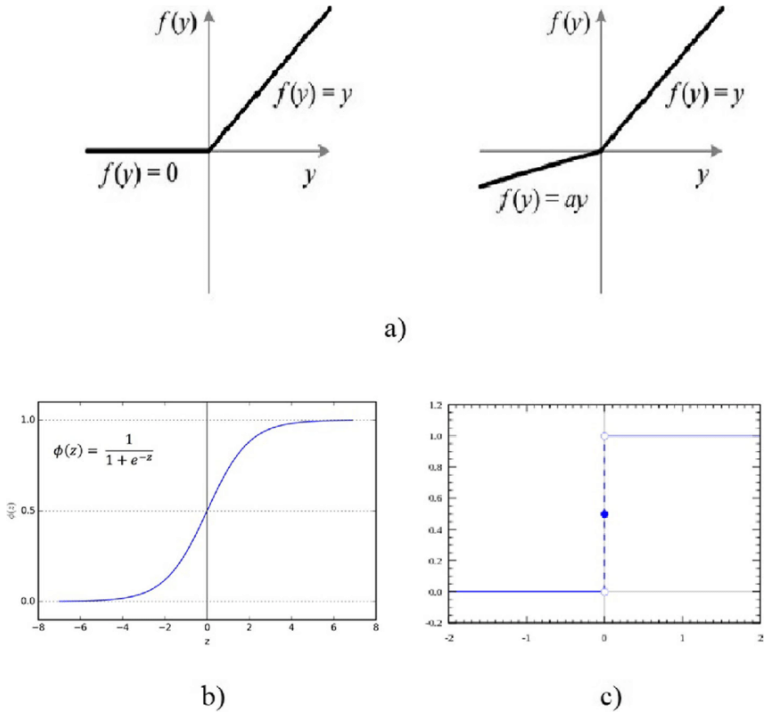


Figura 2.15: Diferentes tipos de funciones de activación comúnmente utilizadas. Arriba vemos la función ReLU (izquierda) y LeakyReLU (derecha). Abajo la función sigmoide (izquierda) y la función escalón (derecha). Fuente: https://www.researchgate.net/figure/Different-Activation-Functions-a-ReLU-and-Leaky-ReLU-37-b-Sigmoid-Activation-Function_fig3_339905203

- **Función escalón**: Esta función asigna un valor 0 a los valores negativos y un valor 1 a los positivos tal y como observamos en su definición en la ecuación 2.27 y en la figura 2.15(c), aunque este umbral se puede modificar. Son funciones adecuadas para problemas binarios.

$$H(x) = \begin{cases} 0, & \text{si } x \leq 0 \\ 1, & \text{si } x > 0 \end{cases} \tag{2.27}$$

Características de las Redes Neuronales

Arquitectura y tipos Las neuronas están típicamente organizadas en múltiples capas. Las neuronas de una capa se conectan con las neuronas de las capas inmediatamente anterior e inmediatamente posterior. La capa que recibe datos externos es la capa de entrada. La capa que produce el resultado final es la capa de salida. Entre ellas existen (o no) más capas ocultas. Entre dos capas, son posibles múltiples patrones de conexión. Pueden estar totalmente conectadas, con cada neurona en una capa conectada a cada neurona en la siguiente capa. Pueden estar agrupados, donde un grupo de neuronas en una capa se conecta a una sola neurona en la siguiente capa, reduciendo así la cantidad de neuronas en esa capa. Las neuronas que únicamente poseen este tipo de conexiones forman un grafo acíclico dirigido y se conocen como redes hacia delante o *feed-forward*, un ejemplo de la arquitectura de este tipo de redes se aprecia en la figura 2.16. Alternativamente, las redes que permiten conexiones entre neuronas en la misma capa o en capas anteriores se conocen como redes neuronales recurrentes. Otro tipo de red neuronal muy utilizado es las redes neuronales convolucionales, muy típicas en el procesamiento de imágenes y con una arquitectura peculiar.

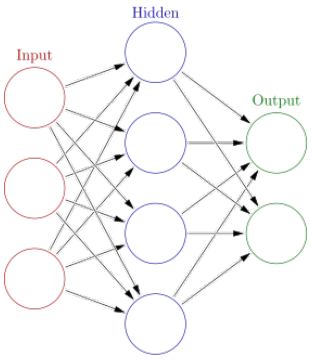


Figura 2.16: Representación de la arquitectura de una red feed-forward con una capa oculta, donde cada nodo circular representa una neurona artificial y cada flecha representa una conexión desde la salida de una neurona artificial a la entrada de otra. Fuente: https://en.wikipedia.org/wiki/Artificial_neural_network.

Función de coste La función de pérdida o función de coste, tiene en cuenta las probabilidades o la incertidumbre de una predicción en función de cuánto varía la predicción respecto del valor real. Esto nos da una visión más del rendimiento del modelo. La pérdida total o *total loss* es la suma de los errores cometidos para cada muestra en conjuntos de

entrenamiento o validación. La pérdida se usa a menudo en el proceso de entrenamiento para encontrar los valores más adecuados de ciertos parámetros del modelo (por ejemplo, pesos en la red neuronal). Durante el proceso de entrenamiento el objetivo es minimizar este valor. Las funciones de pérdida más comunes son la pérdida logarítmica y la de entropía cruzada, así como el error cuadrático medio y las funciones de verosimilitud.

Algoritmo de Descenso del Gradiente El descenso de gradiente es un algoritmo de optimización iterativo que se utiliza generalmente en el aprendizaje automático para minimizar una función de pérdida. La cual describe como de bien funcionará el modelo dado el conjunto actual de parámetros (pesos y sesgos). El descenso de gradiente se utiliza para encontrar el conjunto de parámetros más adecuado. Esto se logra tomando la derivada parcial en un punto dado y luego recorriendo iterativamente el espacio de búsqueda en la dirección negativa de la función gradiente, por tanto decreciente en la función original. A medida que mejora la función de pérdida, los parámetros de un modelo (pesos) se actualizan hasta llegar al punto óptimo que es el mínimo de la función de pérdida (los pesos se actualizan en proporción a la derivada del error). Los dos aspectos clave del descenso de gradiente son la dirección de movimiento y la tasa de aprendizaje o *learning rate*, que se refiere al tamaño del paso y es un hiperparámetro importante en el entrenamiento de las redes neuronales. Existen diversas variaciones de este algoritmo según el tipo de problema.

Algoritmo de Retro-propagación Para calcular el gradiente de la función de coste en las redes neuronales de utiliza el algoritmo de retropropagación o *backpropagation*. Al entrenar una red neuronal, la retropropagación calcula el gradiente de la función de pérdida con respecto a los pesos de la red para un solo ejemplo de entrada-salida, y lo hace de manera eficiente, a diferencia de un cálculo directo del gradiente con respecto a cada peso individualmente. El algoritmo de retropropagación funciona calculando el gradiente de la función de pérdida con respecto a cada peso por la regla de la cadena, calculando el gradiente de una capa cada vez, iterando hacia atrás desde la última capa para evitar cálculos redundantes de términos intermedios en la regla de la cadena. Este eficiente método hace factible el uso del algoritmo de descenso del gradiente de manera acoplada para entrenar redes de varias capas, actualizando pesos para minimizar la función de perdida.

Aunque el algoritmo tiene su origen en las redes *feed-forward* existen generalizaciones para otros tipos de redes. Este algoritmo es esencial para convertir el bucle de optimización en un cálculo tensorial y, en última instancia, permitir a los ordenadores entrenar a la red neuronal en un tiempo razonable.

Configuración utilizada en la Investigación En esta investigación, se ha hecho uso de una red neuronal *feed-forward* con 2 capas de neuronas ocultas y totalmente conectadas, aunque incluyendo ciertas variaciones que comentaremos más adelante. Además el método utilizado para el descenso de gradiente es el algoritmo ADAM, una variación estocástica del original que es muy utilizado actualmente. Sobre la función de coste utilizada y demás parámetros de la red se ahondará en el capítulo 3.

Aprendizaje Profundo Nuestro trabajo no utiliza una técnica de aprendizaje profundo o *deep learning*, ya que para ello la red debería de tener 10 o más capas. Las redes neuronales con un gran número de capas suelen tener una arquitectura que comprime y descomprime la información en su interior de tal manera que la interpretación humana se vuelve extremadamente difícil. Entre sus aplicaciones se incluyen el reconocimiento de imágenes y reconocimiento de voz. Sin embargo, en el campo que aborda nuestra investigación, normalmente queremos evitar las redes profundas, ya que actualmente no hay forma de obtener información adicional sobre los procesos fundamentales que se llevan a cabo en ellas. Las redes profundas son extremadamente útiles para aplicaciones de ingeniería, donde el principal interés es que la predicción sea correcta, no siendo necesario conocer cómo se lleva a cabo.

Teoría de Redes Neuronales

La importancia de las redes neuronales reside fundamentalmente en un resultado teórico denominado «Propiedad de Aproximación Universal de Funciones». A pesar de que la intención de este trabajo no es describir con demasiado detalle la teoría subyacente a las redes neuronales, enunciaremos el resultado junto a algunas definiciones para entender la importancia del mismo. Se obviarán algunas definiciones propias de análisis funcional, que pueden ser consultados en [53].

Definición 2.1 *Sea χ un espacio de funciones y X e Y, dos conjuntos cualesquiera. Se denomina **arquitectura** sobre χ al par $(\mathfrak{F}, p)$ formado por un conjunto de conjuntos de funciones $\mathfrak{F}$ entre X e Y y una función parcial $p : \cup_{J\in\mathbb{N}}\mathfrak{F}^J \to \chi$ que satisface la siguiente condición no trivial:*

Existe algún $f \in \chi$,$J \in \mathbb{N}$ y $f_1, \ldots, f_J \in \mathfrak{F}$ que verifican:

$$f = p\Big(f_1, \ldots, f_J\Big) = p\Big((f_i)_{i=1}^J\Big) \tag{2.28}$$

El conjunto de funciones $f \in \chi$ que satisfacen la relación 2.28 se denota por $NN^{(\mathfrak{F},p)}$.

Pese a la aparente dificultad de la definición, la interpretación de lo que una arquitectura significa es simple, cada elemento de la arquitectura es una función que puede ser descrita como *N* funciones bien definidas en todo momento y actuando paralelamente.

Definición 2.2 *(**Propiedad de aproximación universal**) Una arquitectura definida sobre un espacio funcional χ se dice que posee la propiedad de aproximación universal si $NN^{(\mathfrak{F},p)}$ es un conjunto denso en χ.*

Simplificando, si χ es un espacio de *Banach* con norma $||\cdot||$, una arquitectura sobre χ posee la propiedad de la aproximación universal de funciones si $\forall f \in \chi$, $\forall \varepsilon > 0$, existe una realización de una red neuronal f_{NN} tal que $||f - f_{NN}|| < \varepsilon$.

La propiedad de aproximación universal, UAP (*Universal Approximation Property* en inglés), nos permite caracterizar: las funciones que tratamos de aproximar mediante las redes neuronales (en el espacio χ) y cuando se entiende que una red neuronal aproxima bien dicha función.

Teorema 2.4 *(De **aproximación universal** [54]) Sea $\chi = C(\mathbb{R}^m, \mathbb{R}^n)$ el espacio de funciones continuas, acotadas y de soporte compacto. Sea $(\mathfrak{F}, p)$ la arquitectura de las redes neuronales profundas sobre χ con función de activación σ, que se supone continua.*

Bajo estas condiciones, σ no es una función polinómica sí y solo sí la arquitectura de las redes neuronales profundas posee la UAP.

La consecuencia efectiva del teorema 2.4 es que para cualquier función en $C(\mathbb{R}^m, \mathbb{R}^n)$, existe una red neuronal profunda con función de activación no polinómica y una realización de la red no especificada que la aproxima arbitrariamente bien.

Este resultado recoge a su vez la necesidad de emplear muchas de las características de las redes neuronales mencionadas durante la sección. El ejemplo más claro es la exigencia de al menos una capa oculta en la red neuronal para obtener la aproximación deseada.

La necesidad de que σ no sea polinómica justifica el uso de todas las funciones de activación continuas empleadas. Para los casos no continuos, como las funciones escalón, su uso se justifica según las aplicaciones concretas (problemas binarios).

2.4 Redes Neuronales Bayesianas (BNN)

El uso de las redes neuronales ha provocado una revolución en el *machine learning*, dotando de soluciones a problemas que tradicionalmente eran muy difíciles de resolver. A pesar de esto, los modelos de redes neuronales son propensos al sobre-entrenamiento (se ajustan al ruido subyacente de los datos interpretando erróneamente los mismos), lo cual afecta negativamente a sus capacidades de generalización [55]. Del mismo modo, tienden a mostrar un exceso de confianza en sus predicciones cuando ofrecen intervalos de confianza. Todo esto puede llegar a ser problemático si lo aplicamos a ciertos campos donde pequeños fallos pueden conllevar terribles consecuencias, como por ejemplo la conducción autónoma, los diagnósticos médicos o las finanzas. Es por esto que varios propuestas se han alzado para evitar esta problemática. Entre ellas se encuentra el paradigma Bayesiano, el cual ofrece un riguroso marco para analizar y entrenar redes neuronales consecuentes con sus incertidumbres, y de forma más general, para apoyar el desarrollo de los algoritmos de aprendizaje.

2.4.1 Estadística Bayesiana

Como se explica en [56] este paradigma Bayesiano, en estadística, se contrapone al paradigma frecuentista mayormente en el área de contraste de hipótesis [57]. Se basa en dos ideas sencillas. La primera es que la probabilidad es una medida de la creencia de que ocurra un evento, más que el límite en la frecuencia de ocurrencia cuando la muestra es infinita, que es lo que asume el paradigma frecuentista. La segunda idea es que las

creencias anteriores al evento influyen en las posteriores. En esencia el acercamiento Bayesiano identifica una relación entre las probabilidades condicionadas. El **Teorema de Bayes** representado en la expresión de la ecuación 2.29 resume esta interpretación.

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} = \frac{P(B|A_i)P(A_i)}{\sum_{i=1}^{n} P(B|A_i)P(A_i)} \tag{2.29}$$

En esta ecuación general $\{A_1, \ldots, A_n\}$ forma una partición del espacio muestral, y cada A_i y B representan sucesos estocásticos. El teorema ofrece una manera de determinar la probabilidad de A_i supuesto que B ha ocurrido. Para colocar este teorema en el contexto de las Redes Neuronales Bayesianas, A_i y B serán remplazados con la hipótesis H y los datos D, como ilustra la ecuación 2.30.

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)} = \frac{P(D \cap H)}{P(D)} \tag{2.30}$$

El propósito de la ecuación es entonces, determinar la probabilidad de que los datos apoyen la hipótesis $P(H|D)$. En el futuro esta hipótesis será una distribución predictiva. Por otro lado, $P(D|H)$ representa la probabilidad de que los datos provengan de esta hipótesis, lo cual se multiplica por la creencia de que esta hipótesis sea cierta $P(H)$. Este término se divide entonces por el factor de normalización $P(D)$, a menudo denominado como la evidencia.

Desde una perspectiva informal, podríamos decir que la verosimilitud de la hipótesis H cuando es actualizada mediante nuevos datos D puede conducir a una verosimilitud mejorada a posteriori $P(H|D)$. En esencia, esto nos traduce de manera adecuada el teorema de Bayes a las técnicas de *machine learning*, donde un gran número de ejemplos se usan para actualizar en entendimiento del fenómeno subyacente.

Sin embargo, estas ideas sobre una hipótesis y los respectivos datos no conducen directamente a ningún tipo de propuesta de modelo de *machine learning*. Para ello habrá que desarrollar varios conceptos hasta llegar a la aplicación práctica final.

2.4.2 Implementación en el modelo

Siguiendo la introducción de estadística bayesiana aplicada a redes neuronales, para su implementación en un modelo habríamos de tomar la hipótesis H como una distribución de pesos posibles para la red neuronal, tal y como se realiza en [58].

Denotemos por X a los datos y $p(X|w)$ a la función de probabilidad de un modelo, con $w \in \mathcal{W}$ el vector de parámetros del modelo a estimar. Sea $p(w)$ la estimación anterior o *prior*. La inferencia Bayesiana codifica toda la información disponible sobre el parámetro del modelo w en su distribución posterior, con densidad representada en la ecuación 2.31.

$$p(w|X) = \frac{p(X \cap w)}{p(X)} = \frac{p(w)p(X|w)}{p(X)} \propto p(w)p(X|w) \tag{2.31}$$

En esta ecuación $p(w)$ es el llamado **anterior o *prior*** sobre los pesos w. Es el parámetro más controvertido ya que tenemos que elegirlo a mano y afectará a la distribución posterior que obtengamos. La **verosimilitud** $p(X|w)$ se trata de la probabilidad de que los datos provengan de unos determinados pesos w. La **probabilidad marginal o evidencia** $p(X)$, que se calcula como $p(X) = \int_{\mathcal{W}} p(w)p(X|w)dw$, es un valor constante para un conjunto de datos fijo y, por lo general, es difícil de encontrar. Si lo conociéramos podríamos determinar el posterior ya que podemos evaluar la verosimilitud y el anterior. La notación $\propto$ se refiere a proporcionalidad con una constante de normalización que es independiente del parámetro del modelo w. En la mayoría de las derivaciones Bayesianas esta constante puede ser ignorada sin repercusión alguna.

La inferencia Bayesiana generalmente requiere de computar valores esperados con respecto a la distribución posterior. Por ejemplo, la media posterior, a menudo utilizada para estimación puntual, es un valor esperado de w con respecto a la distribución posterior $p(w|X)$. Sin embargo, a menudo es complicado calcular estos valores esperados, en parte porque la densidad $p(w|X)$ en sí es dificil de manejar ya que la constante normalizadora p(X) a menudo es desconocida. De este modo, para muchas aplicaciones la inferencia Bayesiana se lleva a cabo mediante simulaciones de Monte Carlo con Cade-

nas de Markov (MCMC), lo cual sirve para estimar los valores esperados con respecto a $p(w|X)$ muestreando de esta. Para otras aplicaciones donde w tiene muchas dimensiones o donde el cálculo rápido es de alto interés, el campo de la inferencia variacional es una atractiva alternativa de MCMC. Este caso es el de las redes neuronales donde w representará la distribución de pesos de la red que a menudo es de grandes dimensiones y donde la eficacia computacional es un factor primario.

De ahora en adelante denominaremos los datos de entrada como $\mathbf{x}$ y los datos de salida como $\mathbf{y}$.

Consideremos una tarea sencilla de regresión en una dimensión, a saber: determinar la relación entre la variable de entrada $\mathbf{x}$ y la variable objetivo $\mathbf{y}$. En vez de usar las técnicas tradicionales de estimación máximo verosímil (MLE) o máximo a *posteriori* (MAP) para realizar estimaciones puntuales, podríamos tratar de encontrar la **distribución de los pesos a posteriori** $p(\mathbf{w}|\mathbf{x},\mathbf{y})$ desarrollando de este modo una red neuronal estocástica. Vemos cómo se usa la misma notación que en estadística Bayesiana, $\mathbf{w}$ representa los pesos de la red y $p(\mathbf{w}|\mathbf{x},\mathbf{y})$ se refiere a la distribución de los pesos de la red teniendo en cuenta los datos de entrenamiento para las variables x e y. Entonces la recién encontrada distribución de pesos del modelo induce una distribución de la predicción de la media μ del modelo, tal y como vemos en la ecuación 2.32. Entonces $M(\mathbf{x})$ representa la media de la variable objetivo $\mathbf{y}$ tras haber integrado sobre todo el espacio de pesos.

$$M(\mathbf{x}) = \int \mu(\mathbf{x},\mathbf{w}) p(\mathbf{w}|\mathbf{x},\mathbf{y}) d\mathbf{w} \tag{2.32}$$

Teniendo esto en mente, para predecir adecuadamente el valor de la variable $\mathbf{y}$ para la variable de entrada $\mathbf{x}$, necesitamos muestrear un conjunto de pesos provenientes de la distribución $p(\mathbf{w}|\mathbf{x},\mathbf{y})$ previamente entrenada, los cuales se utilizan para predecir los valores de $\mathbf{y}$ como aparece en la ecuación 2.33. Sin embargo, veremos en la siguiente sección que este método se sustituirá usando inferencia variacional.

$$p(\mathbf{y}|\mathbf{x},\mathbf{w}) = \mathcal{N}(M(\mathbf{x}), \sigma^2(x)) \tag{2.33}$$

2.4.3 Inferencia variacional

En realidad, debido a la naturaleza de caja negra de las redes neuronales, la distribución de los pesos $p(\mathbf{w}|\mathbf{x},\mathbf{y})$ es imposible de calcular de manera exacta, por ello necesitamos inferir una distribución $q(\mathbf{w}|\theta,\mathbf{x},\mathbf{y})$ que la aproxime, donde θ será el conjunto de parámetros de la distribución q. Una manera usual de encontrar esta distribución es mediante inferencia variacional. En ella el proceso de inferencia se toma como un problema de optimización donde las características de distribuciones conocidas son comparadas para encontrar la que mejor se ajuste. Es decir, se trata de encontrar una distribución q, perteneciente a un conjunto de distribuciones $\mathcal{Q}$ que podemos manejar y que sea lo más similar posible a la distribución objetivo p la cual no es fácil de manejar. La optimización gira en torno a la divergencia de Kullback-Leibler (*KL-Divergence*), con raíces en teoría de la información y cuantifica la diferencia en información entre dos distribuciones. Se define como se muestra en la ecuación 2.34.

$$KL(q||p) = \sum_x q(x) ln \frac{q(x)}{p(x)} \tag{2.34}$$

Este valor se aproximará al 0 cuanto más similares sean las distribuciones, teniendo en cuenta que $KL(q||p) = 0$ sí y solo sí $q = p$. En ese caso el desafío es reducir el término KL lo máximo posible. Notar que dada la definición de la divergencia KL en la ecuación 2.34 y por la naturaleza del logaritmo si $q \neq p$ entonces $KL(q||p) \neq KL(p||q)$.

Queremos encontrar una distribución de w (dependiente de un conjunto de parámetros θ) que aproxime la distribución de pesos posterior, es decir, $q(\mathbf{w}|\theta) \sim p(\mathbf{w}|\mathbf{x},\mathbf{y})$. En este caso aproximar se traduce en minimizar la divergencia KL entre las dos distribuciones, tal y como se expresa en la ecuación 2.35 donde se utiliza la esperanza $\mathbb{E}$ según la distribución aproximada de pesos $q(\mathbf{w}|\theta)$.

$$\arg\min_\theta KL[q(\mathbf{w}|\theta)||p(\mathbf{w} \mid \mathbf{x},\mathbf{y})] = \arg\min_\theta -\mathbb{E}_{q(\mathbf{w}|\theta)} \left[ln \frac{p(\mathbf{w} \mid \mathbf{x},\mathbf{y})}{q(\mathbf{w}|\theta)} \right] \tag{2.35}$$

Como ejemplo, la distribución normal que mejor aproxima la distribución de pesos se encuentra tomando $q(\mathbf{w}|\theta) = \mathcal{N}(\mathbf{w}|\mu,\Sigma)$ con $\theta = (\mu,\Sigma)$ para minimizar la divergencia

KL con respecto a θ. Sin embargo, para calcular la divergencia KL necesitamos conocimientos de la distribución posterior de pesos $p(\mathbf{w}|\mathbf{x},\mathbf{y})$, la cual no es posible calcular. Por fortuna, estudios en inferencia variacional [59] nos ofrecen un resultado que nos ayuda a resolver este problema, se puede observar en la ecuación 2.36, donde el termino $-J(q(\theta))$ se llama evidencia del límite inferior (ELBO) ya que se trata de un límite inferior para la probabilidad marginal (o evidencia) $p(\mathbf{y}|\mathbf{x})$.

$$ln\, p(y \mid x) = \overbrace{-\mathbb{E}_{q(\mathbf{w}|\theta)}\left[ln\frac{p(\mathbf{y} \mid \mathbf{x},\mathbf{w})p(\mathbf{w})}{q(\mathbf{w}|\theta)}\right]}^{J(q(\theta))} \overbrace{-\mathbb{E}_{q(\mathbf{w}|\theta)}\left[ln\frac{p(\mathbf{w} \mid \mathbf{x},\mathbf{y})}{q(\mathbf{w}|\theta)}\right]}^{KL[q(\mathbf{w}|\theta)||p(\mathbf{w}|\mathbf{x},\mathbf{y})]} \tag{2.36}$$

La clave de esta ecuación es que la divergencia KL combinada con el término ELBO permanece constante independientemente de la distribución $q(\mathbf{w}|\theta)$. Por lo cual podemos afirmar que la divergencia KL puede ser minimizada maximizando el ELBO o minimizando $J(q(\theta))$ con respecto a θ, resultando la ecuación 2.37. Eso es cierto debido a que el término

$$\arg\min_{\theta} KL[q(\mathbf{w}|\theta)||p(\mathbf{w} \mid \mathbf{x},\mathbf{y})] = \arg\min_{\theta} J(q(\theta)) \tag{2.37}$$

Simplificando el término ELBO obtenemos la siguiente ecuación:

$$\begin{aligned} J(q(\theta)) &= \mathbb{E}_{q(\mathbf{w}|\theta)}\left[-ln\, p(\mathbf{y} \mid \mathbf{x},\mathbf{w}) - ln\,\frac{p(\mathbf{w})}{q(\mathbf{w}|\theta)}\right] \\ &= -\mathbb{E}_{q(\mathbf{w}|\theta)}\left[ln\, p(\mathbf{y} \mid \mathbf{x},\mathbf{w})\right] - \mathbb{E}_{q(\mathbf{w}|\theta)}\left[ln\,\frac{p(\mathbf{w})}{q(\mathbf{w}|\theta)}\right] \\ &= -\mathbb{E}_{q(\mathbf{w}|\theta)}\left[ln\, p(\mathbf{y} \mid \mathbf{x},\mathbf{w})\right] + KL[q(\mathbf{w}|\theta)||p(\mathbf{w})] \end{aligned} \tag{2.38}$$

Es de notar que la divergencia KL en esta ecuación, es la divergencia entre el *prior* $p(\mathbf{w})$ y nuestra distribución aproximada $q(\mathbf{w}|\theta)$ mientras que en la ecuación 2.36 teníamos la divergencia entre el posterior $p(\mathbf{w}|\mathbf{x},\mathbf{y})$ y la distribución aproximada $q(\mathbf{w}|\theta)$. El motivo por el cual hemos llevado a cabo este cambio es que la divergencia KL de la ecuación 2.38 es un cálculo que podemos realizar porque conocemos las distribuciones implicadas, sin embargo en la divergencia KL de la ecuación 2.36 necesitamos la distribución posterior de pesos $p(\mathbf{w}|\mathbf{x},\mathbf{y})$, la cual sabemos que es imposible de calcular. La expresión final de la divergencia KL nos ofrece una relación entre el *prior* y la distribución aproximada de los pesos.

Aunque la elección tanto del *prior* de los pesos $p(\mathbf{w})$ como del tipo de distribución $q(\mathbf{w}|\theta)$ no está cerrada, en la práctica las distribuciones normales se usan para ambos, ya que permite simplificar los cálculos.

Para implementar este método en el modelo, todavía falta hacer algunos cambios. Puesto que la distribución se propaga a lo largo del modelo, la distribución a posteriori para el valor objetivo *y* (predicción) con el enfoque de la inferencia variacional se describe como en la ecuación 2.39.

$$p(\mathbf{y}|\mathbf{x},\mathbf{w}) = \int p(\mathbf{y}|\mathbf{x},\mathbf{w},\theta)\, q(\mathbf{w}|\theta)\, d\theta \tag{2.39}$$

Sin embargo, para trasladar todos estos conocimientos a la red neuronal y su entrenamiento necesitamos definir una función de coste. Tomando el razonamiento seguido con la ecuación 2.38, la función de coste resulta como en la ecuación 2.40, se trata del propio término $J(q(\theta))$. La tarea de la red neuronal será minimizar esta expresión. En primer lugar, maximizar el término $\mathbb{E}_{q(\mathbf{w}|\theta)}\left[ln\, p(\mathbf{y} \mid \mathbf{x},\mathbf{w},\theta)\right]$, por lo que buscamos la estimación máximo verosímil de θ en la distribución posterior de la predicción $p(\mathbf{y}|\mathbf{x},\mathbf{w},\theta)$ desde el punto de vista de la distribución aproximada $q(\mathbf{w}|\theta)$. En segundo lugar queremos minimizar la divergencia KL entre el *prior* $p(\mathbf{w})$ y la distribución que aproximamos $q(\mathbf{w}|\theta)$.

$$loss(\theta) = -\mathbb{E}_{q(\mathbf{w}|\theta)}\left[ln\, p(\mathbf{y} \mid \mathbf{x},\mathbf{w},\theta)\right] + KL[q(\mathbf{w}|\theta)||p(\mathbf{w})] \tag{2.40}$$

La optimización trata de modificar el conjunto de parámetros θ de la distribución q hasta que se asemeje a la distribución posterior real p, en lugar de tratar de modificar la p posterior para que coincida con alguna distribución esperada q.

La conclusión que podemos extraer es que el rendimiento del modelo permanece dependiente de alguna manera al muestreo de $q(\mathbf{w}|\theta)$ durante la evaluación de la red neuronal. Esto supone un problema en cuanto el tiempo y los recursos que consume la red al ejecutarse, que aumentan sensiblemente a causa del muestreo, y hace imposible aplicar la red en procesos a tiempo real.

2.5 *Prior* de Contraste con Ruido (NCP)

Tal y como se menciona al final de la sección anterior, el muestreo durante el entrenamiento de la red supone el mayor obstáculo en la implementación de los métodos Bayesianos en las redes neuronales, ya que reduce la velocidad de computación en al menos en un orden de magnitud. Para evitar este problema debemos encontrar otra manera de llevar a cabo la hipótesis descrita en la sección 2.4.2.

Por fortuna, un reciente estudio en redes neuronales Bayesianas [60] ofrece una perspectiva mejorada del asunto. Básicamente, hace uso de ruido simulado en los valores de entrada y salida para mover la hipótesis desde la distribución de probabilidad de los pesos de la red neuronal, hasta una distribución de probabilidad en el espacio de datos. Esta simulación de ruido no es otra cosa que una manera de aumentar los datos y es vital para poder aplicar inferencia variacional en el espacio de datos. Además este *Prior* de Contraste con Ruido (NCP de *Noise Contrastive Prior* en inglés) también mejora la detección de regiones fuera de la distribución o regiones OOD (*out of distribution*). En definitiva, este enfoque permite un cálculo aproximado de la distribución de salida sin necesidad de muestrear los pesos para llegar a una estimación confiable de la incertidumbre.

En el caso de una regresión, el NCP se puede describir como una combinación de un ***prior* de entrada** $p(\mathbf{x})$ y un *prior* de salida dado el de entrada $p(\mathbf{y}|\mathbf{x})$. Esta relación se especifica como $p(\mathbf{x},\mathbf{y}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x})$. En la práctica el *prior* de entrada se usa a menudo para describir la distribución de valores de entrada $\tilde{\mathbf{x}}$ generados en regiones OOD. Esto se realiza añadiendo a los datos existentes un ruido Gaussiano aleatorio $\sim \mathcal{N}(0, \sigma_x^2)$, tal y como se expresa en la ecuación 2.41, donde $\mathbf{x_i}$ son los datos de entrada y σ_x^2 es un hiperparámetro que nos indica como de lejos queremos que se muestreen los puntos OOD.

$$p_{prior}(\tilde{\mathbf{x}}) = \frac{1}{N}\sum_{i}^{N} \mathcal{N}(\tilde{\mathbf{x}} - \mathbf{x_i}|0, \sigma_x^2) \tag{2.41}$$

Dar con el valor adecuado de σ_x^2 puede ser una tarea delicada, ya que si el valor es demasiado pequeño esto puede derivar en predicciones con demasiada poca incertidumbre,

mientras que los valores demasiado grandes disminuirán la capacidad del modelo para detectar regiones OOD o fuera de la distribución. Sin embargo, en la experimentación llevada a cabo en [60], quedó claro que la elección de σ_x^2 era bastante robusta dentro de unos límites razonables.

Por otro lado, el objetivo del *prior* de salida es predecir el valor correcto de $\mathbf{y}$ para el valor de entrada $\mathbf{x}$, pero también para el valor de entrada con ruido $\tilde{\mathbf{x}}$. Al mismo tiempo, es deseable que el *prior* de salida tenga una entropía relativamente alta de forma que represente una mayor incertidumbre en relación con los valores de entrada en la región OOD. En la ecuación 2.42 podemos ver como la definición de **prior de salida** sigue estas características. Apreciamos como la distribución se centra alrededor del objetivo original, con cierta variación de acuerdo al hiper-parámetro σ_y^2 el cual debería reflejar la alta incertidumbre del prior en el caso de valores de entrada OOD.

$$p_{prior}(\tilde{\mathbf{y}}|\tilde{\mathbf{x}}) = \mathcal{N}(\mathbf{y}, \sigma_y^2) \tag{2.42}$$

Teniendo estos nuevos *priors* en mente, el cambio hacia el espacio de datos puede explorarse más detenidamente. Aunque el objetivo continúa siendo encontrar una distribución predictiva de la variable objetivo $\mathbf{y}$ (tal como se expresa en la sección 2.4.2) el proceso será alterado. En vez de optimizar la distribución de pesos de $q(\mathbf{w}|\theta)$ comparado con el *prior* en los pesos $p(\mathbf{w})$, ahora la distribución de la media $q(\mu|\mathbf{x},\theta)$, inducida por $q(\mathbf{w}|\theta)$, se compara directamente con el *prior* de salida $p(\tilde{\mathbf{y}}|\tilde{\mathbf{x}})$. Esencialmente, para los datos provenientes de las regiones OOD, la incertidumbre epistémica esperada convergerá hacia el *prior* de salida. Dado que para este método solo necesitaremos de la capa de salida lineal en la divergencia KL ($q(\mu|\mathbf{x},\theta)$ se compara directamente con $p(\tilde{\mathbf{y}}|\tilde{\mathbf{x}})$), entonces $q(\mu|\mathbf{x},\theta)$ puede ser analíticamente derivado de los parámetros θ de la distribución $q(\mathbf{w}|\theta)$, sin necesidad de muestreo. Para implementarlo en la práctica en la función de coste, deberemos reparametrizar la divergencia KL (definida originalmente en el espacio de los pesos $\mathbf{w}$) en el espacio de datos. Esta reparametrización se muestra en la ecuación 2.43, donde $p(\mu(\tilde{\mathbf{x}})) = \int \mu(\tilde{\mathbf{x}},\theta)p(\theta)d\theta$ y $q(\mu(\tilde{\mathbf{x}})) = \int \mu(\tilde{\mathbf{x}},\theta)q(\theta)d\theta$ son las distribuciones de la predicción de la media inducida por los pesos. Como resultado, en vez de especificar el

prior en el espacio de pesos, podemos especificarlo en el espacio de salida.

$$\begin{aligned}
\mathbb{E}_{p(\mathbf{x},\mathbf{y})}[ln\, p(\mathbf{y}|\mathbf{x})] &= \mathbb{E}_{p(\mathbf{x},\mathbf{y})}\left[ln \int p(\mathbf{y}|\mathbf{x},\theta)p(\theta)\frac{q(\theta)}{q(\theta)}d\theta\right] \\
&\geq \mathbb{E}_{p(\mathbf{x},\mathbf{y})}\left[q(\theta)\, ln\, p(\mathbf{y}|\mathbf{x},\theta)\frac{p(\theta)}{q(\theta)}d\theta\right] \\
&= \mathbb{E}_{p(\mathbf{x},\mathbf{y})}\left[\mathbb{E}_{q(\theta)}[ln\, p(\mathbf{y}|\mathbf{x},\theta)] - KL[q(\theta)||p(\theta)]\right] \\
&= \mathbb{E}_{p(\mathbf{x},\mathbf{y})}\left[\mathbb{E}_{q(\theta)}[ln\, p(\mathbf{y}|\mathbf{x},\theta)] - \mathbb{E}_{p(\tilde{\mathbf{x}}|\mathbf{x})}\left[KL[q(\theta)||p(\theta)]\right]\right] \\
&\approx \mathbb{E}_{p(\mathbf{x},\mathbf{y})}\left[\mathbb{E}_{q(\theta)}[ln\, p(\mathbf{y}|\mathbf{x},\theta)] - \mathbb{E}_{p(\tilde{\mathbf{x}}|\mathbf{x})}\left[KL[q(\mu(\tilde{\mathbf{x}}))||p(\mu(\tilde{\mathbf{x}})|\mathbf{x})]\right]\right]
\end{aligned} \tag{2.43}$$

Previamente se ha reparametrizado la divergencia KL del espacio de los pesos al espacio de salida o espacio de datos; con el cambio de variables, esto es equivalente si la función $\mu(\cdot,\theta)$ es continua y biyectiva con respecto a θ. Dado que esta afirmación no se sostiene para redes neuronales, ya que varios parámetros pueden conducir a la misma distribución predictiva, utilizamos la aproximación $\approx$.

Implementando estos cambios a la estructura del modelo podemos ofrecer una función de coste actualizada, la vemos en la ecuación 2.44. De nuevo el primer término representa el logaritmo de la verosimilitud en negativo (NLL), mientras que el segundo término compara la diferencia en cuanto información (divergencia KL) entre el *prior* de salida y la distribución de la media del modelo para datos de entrada OOD. El factor γ se utiliza para balancear la importancia relativa de cada término en la suma total.

$$loss(\theta) = -\mathbb{E}_{p(\mathbf{x},\mathbf{y})}\left[ln\, p(\mathbf{y}|\mathbf{x},\theta)\right] + \gamma KL(p_{prior}(\tilde{\mathbf{y}}|\tilde{\mathbf{x}})||q(\mu|\tilde{\mathbf{x}},\theta)) \tag{2.44}$$

Sabemos que si $q \neq p$ entonces $KL(q||p) \neq KL(p||q)$, sin embargo en el caso de NCP cambiar la dirección de la divergencia KL no parece ocasionar cambios significativos según [60].

Como resumen de lo que este modelo adaptado trata de conseguir, en primer lugar nos ofrece una predicción y un intervalo de confianza para la variable $\mathbf{y}$ en las áreas donde hay suficiente información. También incentiva mayores incertidumbres en la zonas fuera de la distribución (OOD) a través de la implementación de los *priors* con ruido $p(\tilde{\mathbf{x}})$ y $p(\tilde{\mathbf{y}}|\tilde{\mathbf{x}})$.

De manera más precisa, la distribución predictiva converge al *prior* de salida dependiendo de la probabilidad de que el punto de entrada esté localizado fuera del área de entrenamiento. A su vez, dado que esta probabilidad se deriva del ruido subyacente a los datos de entrenamiento iniciales, se puede caracterizar como un efecto de límite que identifica la transición entre las regiones donde se encuentran los datos de entrenamiento y las regiones OOD. En la práctica, resulta que a menudo es suficiente identificar estos límites, después de lo cual el efecto se propaga bastante bien hacia el exterior.

Finalmente, dado que todo esto se hace sin ningún muestreo durante la evaluación de la red neuronal, las redes neuronales Bayesianas que emplean NCP no sufren los mismos inconvenientes en cuanto a la velocidad computacional que las redes Bayesianas normales. Lo que convierte a las BNN-NCP en un método ideal para ser aplicado en problemas complejos de modelado en tiempo real.

Una de las principales ventajas de este método BNN-NCP es que consigue separar la incertidumbre debido al modelo o **epistémica**, de la incertidumbre debido al ruido subyacente de los datos o **aleatoria**, esto resultara clave en el desarrollo del proyecto. En los sucesivos apartados se irá aportando más detalles a los conceptos de incertidumbre epistémica y aleatoria.

2.5.1 BNN-NCP Unidimensional

Tras presentar las bases del modelo y detallar su funcionamiento, se muestra un caso práctico en una dimensión que había sido considerado en [61]. Los datos siguen una relación sinusoidal $\mathbf{y} = f(\mathbf{x})$ y se usa el metodo BNN-NCP para mostrar sus ventajas.

Como sabemos en el método BNN-NCP se aleja de las distribuciones en el espacio de pesos y mueve los elementos estocásticos al espacio de datos para evitar la necesidad de simulaciones de Monte Carlo. Además añade un factor un tanto más humano, la idea Bayesiana de los *priores* de entrada y de salida, que representan valores esperados previamente en vez de puros cálculos.

El desarrollo del modelo se centra alrededor de la habilidad de capturar la función de la variable objetivo $\mathbf{y}$. Aunque para este modelo se han usado dos capas de neuronas totalmente conectadas entre sí, la principal diferencia con una red neuronal común se encuentra en las siguientes capas. La aproximación variacional se lleva a cabo para determinar la distribución de medias $\mu(\mathbf{x}, \mathbf{w}, \theta)$ descrita en la ecuación 2.32, y se hace mediante la capa variacional *DenseReparameterizacion* del paquete *TensorFlow Probability*. Este paquete es una variación probabilística del conocido paquete *Tensorflow* tan útil en los cálculos tensoriales de redes neuronales en *Python*, (lenguaje de programación mediante el cual se ha llevado a cabo el trabajo). Adicionalmente, una capa independiente efectúa una estimación puntual de la varianza $\sigma^2(\mathbf{x}, \theta)$ de la distribución de salida $p(\mathbf{y}|\mathbf{x}, \mathbf{w}, \theta)$. Entonces la varianza de $\mu(\mathbf{x}, \mathbf{w}, \theta)$ representa la incertidumbre propia del modelo o epistémica, mientras que la varianza $\sigma^2(\mathbf{x}, \theta)$ nos ofrece la incertidumbre aleatoria o propia de los datos.

Este modelo trata de predecir la variable objetivo $\mathbf{y}$ al igual que una red neuronal común, sin embargo, como sabemos trata de aproximarlo mediante distribuciones, en vez de mediante estimación puntual. Es por eso que la función de coste optimiza el parámetro θ correspondiente a parámetros de una distribución $q(\theta)$ que elegimos y que, en la práctica, acostumbra a ser la distribución normal. De esta manera en vez de obtener predicciones puntuales como salida, obtenemos una distribución de la media. La propia media sería la predicción de la variable objetivo y la desviación estándar sería la incertidumbre propia del modelo o epistémica. Además, a la función de coste se le añade otro término de divergencia KL donde se comparan la varianza estimada de cada punto de salida $\sigma(\mathbf{x}, \theta)$, con un hiper-parámetro que hace referencia al ruido esperado en el valor de salida. De esta manera se consigue que la incertidumbre aleatoria o propia de los datos converja en las regiones OOD a un determinado valor. Este término no afecta negativamente a la predicción en las áreas donde hay suficientes datos debido al término NLL, presente en la ecuación 2.44.

Un efecto muy beneficioso de este método es que incentivar incertidumbres mayores cerca del límite de los datos de entrenamiento causa un efecto de propagación que ase-

gura una alta incertidumbre en las regiones OOD. A su vez, la desventaja del método se encuentra en que requiere cierto conocimiento previo del sistema a modelar ya que usamos la noción de *priors* que se basan en suposiciones a priori. Sin embargo, en física de plasmas el problema a resolver no se encuentra en el conocimiento previo del sistema, sino en la velocidad de computación de los métodos analíticos que existen.

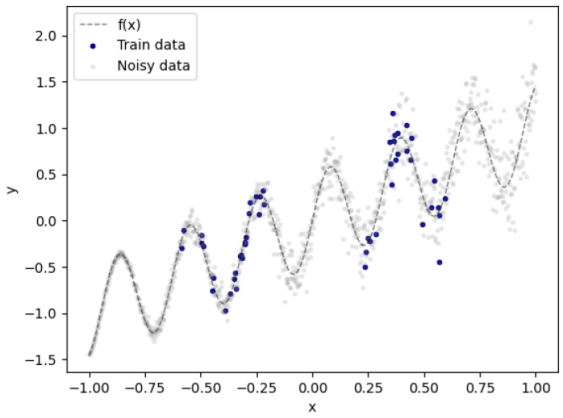


Figura 2.17: Conjunto de datos elegidos (en azul) para probar el método BNN-NCP. Fuente: [61].

Para mostrar la aplicación del método BNN-NCP a una dimensión se genera un conjunto de datos alrededor de la función subyacente $f(\mathbf{x}) = \frac{1}{2}sen(20\mathbf{x}) + \mathbf{x}$. Se introduce ruido según la distribución $\mathcal{N}(0, 0.3(\mathbf{x}+1))$. Luego se muestrean 50 puntos de entrenamiento dentro de $[-0.6, -0.2]$ y $[0.2, 0.6]$. Este será el conjunto de datos de entrenamiento, podemos observarlo en la figura 2.17.

En la figura 2.18 podemos observar el resultado de aplicar este modelo en el conjunto de datos provenientes de una función sinusoidal. Se predicen la variable objetivo tanto en las zonas de entrenamiento como en las regiones fuera de ellas (OOD). Además de la predicción para la variable objetivo, que corresponde con la media de la distribución predictiva, se muestran también los errores epistémico y aleatorio, debido al modelo y a los datos respectivamente. De esta manera se puede distinguir entre regiones con datos de entrenamiento y regiones OOD, siendo apreciable la diferencia en términos de incertidumbre entre ellas. En las zonas donde el modelo tiene datos para aprender, la incertidumbre epistémica mantiene unos valores razonables en torno a la predicción, pero cuando se aleja de estas regiones la incertidumbre epistémica se dispara, dando a entender que la predicción

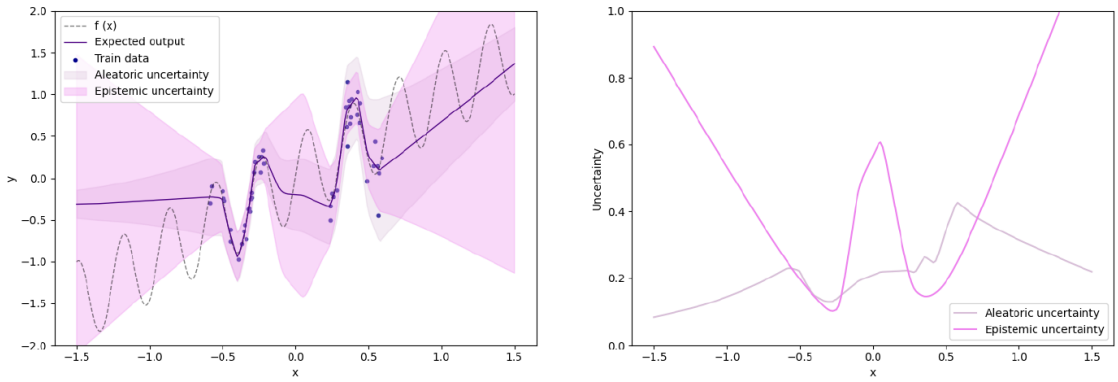


Figura 2.18: Se muestra las predicciones realizadas por la BNN-NCP después de 4000 ciclos de entrenamiento. A la izquierda se muestra la predicción del valor objetivo **y** *con sus incertidumbres. A la derecha el gráfico que representa la incertidumbre aleatoria y epistémica. Fuente: [61].*

de esa región es fruto de la extrapolación y la confianza es baja. De otra manera, la incertidumbre aleatoria es un tanto mayor en las zonas donde existen datos ya que es gracias a estos datos que obtiene el ruido que puede dar lugar a este tipo de error, en las regiones OOD esta incertidumbre desciende sensiblemente, pero también podríamos observar que aumenta al igual que la incertidumbre epistémica, ofreciéndonos el mismo tipo de información.

Capítulo 3

Metodología

3.1 BNN-NCP Multidimensional

En las secciones 2.4 y 2.5, el método de Red Neuronal Bayesiana de Constraste con Ruido (BNN-NCP) ha sido ampliamente detallado, finalizando con el apartado 2.5.1 donde se describía un ejemplo del modelo en un problema unidimensional. Sin embargo, para que el método BNN-NCP sea realmente útil en esta investigación deberemos de extenderlo a una dimensionalidad mayor, ya que para elaborar un modelo sustituto de los modelos EPED y EuroPED (detallados en los apartados 2.2.1 y 2.2.2 respectivamente) tomaremos más de una variable de entrada y de salida.

A nivel de concepto, esto no supone un gran problema debido a que todas las herramientas matemáticas que hemos definido para el método BNN-NCP pueden extenderse a varias variables.

A la hora de definir los *priors* de entrada y salida, resulta más idóneo tomar distribuciones normales unidimensionales para cada variable en vez de una única distribución normal multidimensional, pero como es equivalente a nivel de concepto si lo hiciéramos con la distribución normal multidimensional tendríamos la expresión de la ecuación 3.1 para L dimensiones de entrada. En esta expresión $\tilde{\mathbf{x}}$ y μ_x representan los vectores de valores de entrada con ruido y la media de la distribución (valor de entrada original en la

práctica) respectivamente. El símbolo Σ_x representa la matriz de covarianza donde solo son no nulos los elementos de la diagonal, que corresponden con los hiper-parámetros $\sigma_{x,i}^2$ que nos indican como de lejos de la región de los datos vamos a muestrear los datos OOD. Estos hiper-parámetros, en la práctica, corresponden con la desviación estándar de la normal unidimensional de cada variable.

$$p_{prior}(\tilde{\mathbf{x}}) = \mathcal{N}_L(\mu_x, \Sigma_x) \tag{3.1}$$

$$\Sigma_x = \begin{bmatrix} \sigma_{x,1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{x,2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{x,L}^2 \end{bmatrix}$$

Análogamente, para el *prior* de salida tenemos la ecuación 3.2 para K dimensiones de salida, donde tanto $\tilde{\mathbf{x}}$ como $\tilde{\mathbf{y}}$ representan los vectores de valores de entrada y de salida respectivamente y en la práctica $\mu_y = \mathbf{y}$. De nuevo Σ_y tiene la misma forma que en la ecuación 3.1 solo que esta vez es de tamaño $K \times K$ (en vez de $L \times L$) y los valores $\sigma_{y,j}$ representan los hiper-parámetros de desviación estándar para cada variable de salida, que tal y como se comentó en la sección 2.5, deberían reflejar la alta incertidumbre del prior en el caso de valores de entrada OOD.

$$p_{prior}(\tilde{\mathbf{y}} \mid \tilde{\mathbf{x}}) = \mathcal{N}_K(\mu_y, \Sigma_y) \tag{3.2}$$

$$\Sigma_y = \begin{bmatrix} \sigma_{y,1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{y,2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{y,K}^2 \end{bmatrix}$$

De nuevo, en el caso multidimensional el objetivo continúa siendo mover el término de divergencia KL del espacio de pesos al espacio de datos realizando una aproximación y optimizarlo junto con el término negativo del logaritmo de la verosimilitud (NLL), según el vector de parámetros θ. Siguiendo la estructura de la ecuación 2.44, en el término KL comparamos la distribución $p(\tilde{\mathbf{y}}|\tilde{\mathbf{x}})$ con la distribución $q(\mu_y, \tilde{\mathbf{x}}, \theta)$, siendo en este ca-

so $\tilde{\mathbf{y}}$, $\tilde{\mathbf{x}}$ y μ_y vectores multidimensionales. Por fortuna, en el caso multidimensional esta aproximación converge, ya que la divergencia KL admite la comparación entre dos distribuciones multidimensionales [62]. Sin embargo, como ya se ha adelantado, en la práctica se usan distribuciones unidimensionales equivalentes por simplicidad, por lo cual no sería ni siquiera necesario hacer la aclaración y los terminos KL correspondientes se sumarían en la función de coste.

Si adaptamos la función de coste resultante al método con L variables de entrada y K variables de salida obtenemos la expresión de la ecuación 3.3. Se realiza la suma de los términos para cada variable de salida representada por el índice j.

$$loss(\theta) = \sum_{j}^{K}(-\mathbb{E}_{p(\mathbf{x},y_j)}\left[ln\, p(y_j|\mathbf{x},\theta)\right] + \gamma_{epi}KL(p_{prior}(\tilde{y}_j|\tilde{\mathbf{x}})||q(\mu_{y_j}|\tilde{\mathbf{x}},\theta))+$$

$$+\gamma_{alea}KL(\mathcal{N}(0,\sigma_j^2(\tilde{\mathbf{x}},\theta))||\mathcal{N}(0,\sigma_{y_j}^2))) \tag{3.3}$$

El primer término se denomina NLL (del ingés *Negative Log-Likelihood*) y es el negativo de la esperanza del logaritmo de la distribución a *posteriori* de la predicción. Tomando las L variables de entrada representadas por el vector $\mathbf{x}$ en la ecuación, la salida es la distribución $p(y_j|\mathbf{x},\theta)$ para la variable objetivo y_j con índice j entre las K variables. Mediante la minimización del NLL en función de θ se obtiene el valor máximo verosímil para el vector de parámetros θ. El término correspondiente al error epistémico es el segundo, con el coeficiente γ_{epi} que muestra la importancia relativa en la suma. Se trata de la divergencia KL que tanto se ha comentado, solo que en este caso se comparan la distribución aproximada de salida que ofrece la red con valores en la región OOD como entrada $q(\mu_{y_j}|\tilde{\mathbf{x}},\theta)$, y la distribución del *prior* de salida, es decir, los valores de salida que esperamos que se encuentren fuera de la zona de datos, como se aprecia en la ecuación 3.2. El último término, con su peso relativo γ_{alea}, se trata del correspondiente al error aleatorio, este término provoca que la incertidumbre aleatoria converja adecuadamente para valores razonables en la zona de datos y para valores sensiblemente mayores o menores en las regiones OOD. Se comparan mediante la divergencia KL dos distribuciones normales unidimensionales centradas en el 0, la primera con desviación estándar correspondiente a la varianza estimada por la red para cada punto de salida tomando valores de entrada OOD $\sigma_j^2(\tilde{\mathbf{x}},\theta)$, la

segunda con desviación estándar igual a un hiper-parámetro que hace referencia al ruido esperado en el valor de salida σ_{y_j}, suele coincidir con el error relativo que esperaríamos para esa variable.

3.2 Arquitectura de la red y Implementación de NCP

Para entender completamente el método BNN-NCP y su funcionamiento práctico, en esta sección se describirá en detalle tanto la arquitectura de la red, como el proceso de entrenamiento de la misma.

3.2.1 Arquitectura

En la figura 3.1 podemos apreciar la arquitectura de la red que se ha usado en esta investigación. Se trata de una red neuronal completamente conectada, solo que tiene algunos matices que la convierten en una red neuronal Bayesiana. Se toman L variables de entrada en la capa de entrada y se conectan completamente a la primera capa oculta con N1 neuronas, la cual está a su vez totalmente conectada a una segunda capa oculta con N2 neuronas. Por último, esta última capa se conecta a la capa de salida con K variables. Hasta aquí no existe ninguna diferencia entre esta red neuronal y una red neuronal convencional. Sin embargo, como podemos apreciar en la figura 3.2, cada salida en la red de la figura 3.1 no es realmente una única neurona, si no que se escinde en 3 por cada variable de salida.

Para cada salida y_j tenemos en primer lugar una capa variacional que mediante la función *DenseReparametrization* del módulo *TensorFlow Probability* se obtiene la media de la distribución de salida μ_j. Una capa variacional, básicamente es una capa de la red neuronal que opera mediante distribuciones en vez de mediante puntos. A su vez el valor de salida σ_j se trata de una neurona completamente conectada que nos ofrece la desviación estándar perteneciente al ruido de los datos. Con estos dos valores se construye una distribución normal unidimensional que será muy útil en el entrenamiento. A efectos prácticos, la desviación estándar de esta distribución se corresponde con la incertidumbre aleatoria,

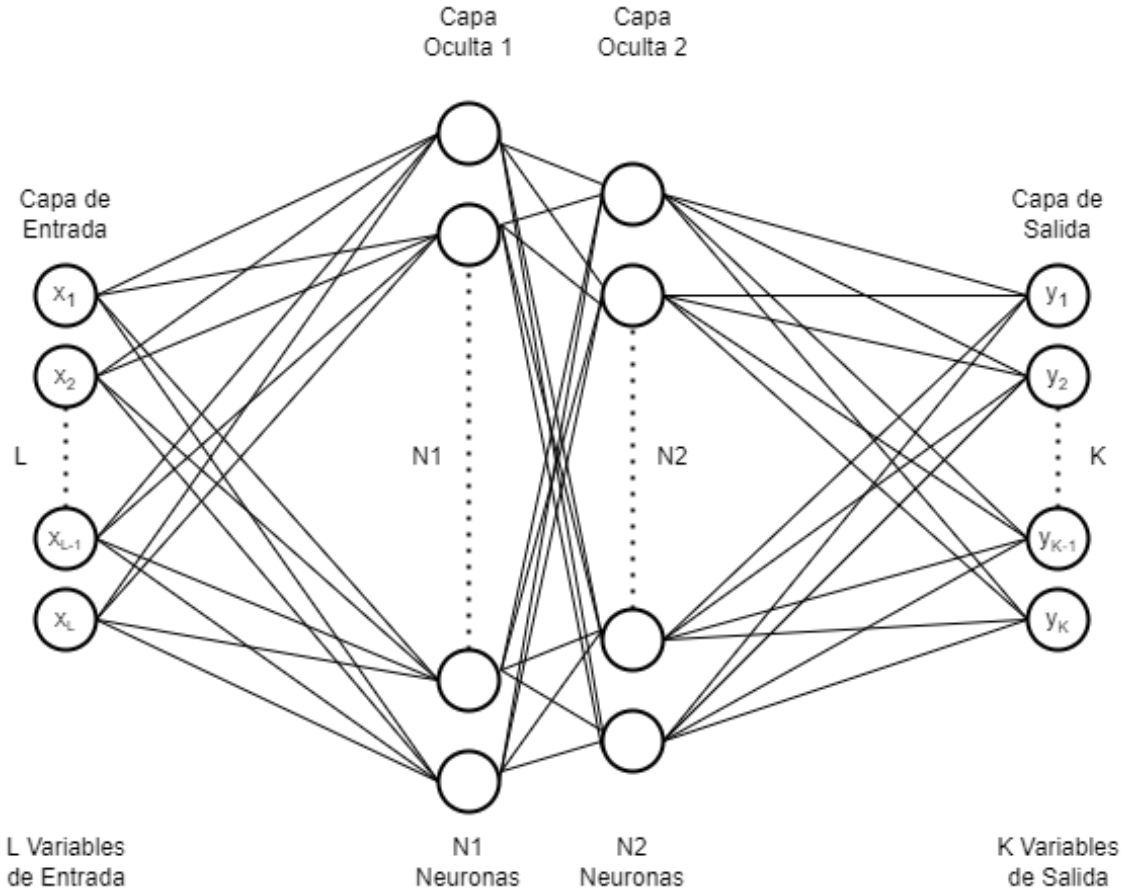


Figura 3.1: Arquitectura de la red neuronal bayesiana con prior de contraste de ruido utilizada a lo largo de la investigación.

es decir, la que es propia del ruido de los datos de entrada. Por último, como vemos en la figura 3.2, la tercera salida se trata de la propia distribución de la media, cuyo valor medio representará la predicción de la red neuronal y su desviación estándar corresponderá con la incertidumbre epistémica. Esta salida utiliza una función auxiliar denominada *mean_dist_fn*, definida en el código (Anexo 5.3). En esta función, se hace uso de la capa variacional para obtener la distribución de la media. Para entender su funcionamiento debemos introducir el concepto de *kernel*. Se trata de las diferentes distribuciones de pesos **w** y sesgos **b** del modelo, y que se alojan a *posteriori* en la capa variacional. Es decir, esquemáticamente se compone de dos matrices (aunque podríamos denominar *kernel* tan solo a la primera), la matriz bidimensional **w** que representa los pesos de las neuronas, donde cada elemento es una distribución con una determinada media y desviación estándar, y la matriz de sesgos unidimensional **b**, donde cada elemento también es una distribución, de este modo en vez de usar escalares como pesos y sesgos, utilizamos distribuciones. Con

esto, en la función *mean_dist_fn* obtenemos del *kernel* de la capa variacional, la media y la desviación estándar de las distribuciones de los pesos **w** y la media de los sesgos **b**, y calculamos una distribución normal de salida que representará la distribución de la media deseada, representada en la figura 3.2. La media de esta distribución corresponderá con $media = E[\mathbf{w}] \cdot d + E[\mathbf{b}]$, siendo d la salida de la última capa de la red neuronal, y la desviación estándar con $desv\_est = \sqrt{d^2 \cdot Var[\mathbf{w}]}$, por lo cual la distribución de la media será $\mathcal{N}(media, desv\_est)$.

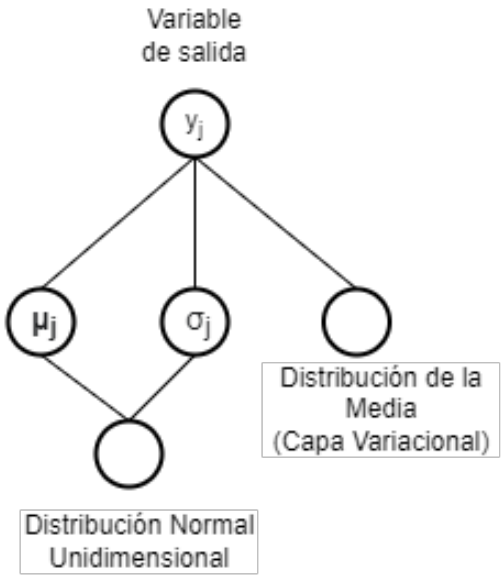


Figura 3.2: Desglose de la capa de salida para cada variable de salida.

3.2.2 NCP y Entrenamiento

A la hora de entrenar el modelo, debemos de proporcionarle los datos de entrada y los datos de salida del conjunto de entrenamiento, el cual tomamos como el 90 % del total de datos. A parte de las columnas pertenecientes a las variables de salida, también necesitaremos una nueva variable llamada $sigma\_y_j$ siendo j el índice de la variable. Se trata del error esperado para cada variable de salida, y se obtiene generalmente calculando la diferencia entre los datos experimentales y los del modelo a sustituir. También necesitaremos especificar un hiper-parámetro $\sigma_{x,i}$ para cada variable de entrada, que representa cómo de lejos de los datos reales queremos muestrear los datos OOD para cada variable de entrada.

En el código, (Anexo 5.3), podemos observar la función *NCP_train_step* que lleva a cabo el proceso de entrenamiento aplicando el método NCP. En primer lugar se toman los datos de salida y_j para cada variable y se ajusta a la forma deseada. Después se modifican los datos de entrada originales para crear los datos OOD. Después de esto, se crea una distribución normal con media el valor y_j y desviación estándar $sigma\_y_j$ para cada variable de salida, que representa la distribución de los valores de salida OOD. Esto sucede para cada entrada del *dataset* de entrenamiento.

Tras esto se llama al modelo y se obtienen las salidas para los datos de entrada originales y para los datos OOD generados, obteniendo cuatro distribuciones por variable de salida: distribución de la media y de ruido para datos originales, y las mismas dos distribuciones para datos OOD. A continuación se calcula el primer término de la ecuación 3.3, para cada variable de salida, se calcula la probabilidad de que la distribución de la media (entrenada) tome el valor de salida real y_j, se aplica el logaritmo y se cambia de signo, dando lugar al término NLL.

Tras esto, es preciso calcular las divergencias KL. Como se muestra en la ecuación 3.3, tendremos 2 divergencias KL por cada variable de salida, la primera correspondiente al error epistémico y la segunda al error aleatorio. El término del error epistémico se calcula comparando las siguientes distribuciones: la primera, la distribución normal que comentábamos hace dos párrafos, con media el valor y_j y desviación estándar σ_y_j, que representa la distribución de los valores de salida OOD, y la segunda, la distribución de salida que obtenemos introduciendo en el modelo los datos de entrada OOD que hemos generado anteriormente.
El término correspondiente al error aleatorio compara dos distribuciones normales con media 0 y desviación estándar, en el caso de la primera, la misma que la distribución del error aleatorio que obtenemos como salida pasando datos OOD al modelo; en la segunda, un valor prefijado que a menudo corresponde con un porcentaje de los datos de salida real (entre un 10 % y un 20 %) o, lo que es lo mismo, un error relativo prefijado.

Notar que, cada una de estas comparaciones entre distribuciones mediante la diver-

gencia KL ocurren para cada entrada del conjunto de entrenamiento, es decir, por cada vector de L valores de entrada tenemos K distribuciones de la media y K distribuciones del ruido. Además tendremos K valores numéricos para cada uno de los términos NLL, KL epistémico y KL aleatorio. Esto ocurre para cada entrada de los datos de entrenamiento, pero el código toma un número igual al hiper-parámetro *batch_size* de entradas en cada iteración. Eso nos da una idea de la complejidad y del número de procesos que se llevan a cabo cuando la red neuronal está en el proceso de aprendizaje.

3.3 Datos JET

Para elaborar esta investigación, se ha hecho uso de una base de datos con casi mil entradas. Esta proviene de datos experimentales con descargas de plasmas en el tokamak JET-ILW (*ITER Like Wall*) localizado en Culham, Reino Unido. Para obtener los datos del pedestal se ha utilizado el ajuste *mtanh*, detallado en el apartado 2.1.2, sobre los datos experimentales. Además se han obtenido las salidas de los códigos EPED y EuroPED, ambos modelos detallados en los apartados 2.2.1 y 2.2.2 respectivamente. La base de datos se describe completamente en [38]. De este modo se incluye los datos de entrada y de salida necesarios para construir los modelos pertinentes.

Tan solo consideraremos algunas de las variables de este *dataset*. Los parámetros que nos interesan son:

- $Ip[MA]$ corriente del tokamak en MA.
- $Bt[T]$ campo magnético toroidal en T.
- $Triang.$ triangularidad del plasma definida en la ecuación 2.5
- $kappa$ elongación del plasma.
- $Ptot[MW]$ potencia total en MW.
- $Gas[1e22/s]$ tasa de suministro de combustible en $10^{22}s^{-1}$.
- Te_ped_exp la temperatura en lo alto del pedestal con los datos de la descarga de plasma en keV.

- *ne_ped_exp* la densidad en lo alto del pedestal con los datos de la descarga de plasma.
- *Te_ped_SC* y *Te_ped_EPED* la predicción de temperatura en lo alto del pedestal según los modelos EuroPED y EPED, respectivamente, en keV.
- *Delta_exp*, *Delta_SC* y *Delta_EPED* la anchura del pedestal experimental, según EuroPED y EPED, respectivamente.
- $neped\_pre\_2$ predicción de la densidad del pedestal según EuroPED.
- *beta_n_exp* y *beta_n_pred* el valor experimental y la predicción de EuroPED de la beta plasmática global, respectivamente.
- $Rmag[M]$ y $rminor[m]$ los correspondientes radios mayor y menor definidos en las ecuaciones 2.3 y 2.4 respectivamente.
- $Zeff\_h$ el número atómico Z efectivo para los iones del plasma.

A través de este conjunto de datos de cerca de 1000 entradas, se podrá entrenar una red neuronal tal y como ha sido descrita a lo largo de esta memoria, de manera que podrá replicar el comportamiento de los pertinentes modelos y arrojar información sobre la física escondida tras este complejo proceso.

Capítulo 4

Resultados

4.1 Primeros pasos

Antes de poder construir un modelo sustituto operativo, necesitamos probar la validez de la BNN-NCP en una dimensionalidad menor. Por ello empezamos extendiendo la red del código original utilizado en el apartado 2.5.1, la cual poseía una dimensión de entrada y otra de salida con sus respectivas incertidumbres.

4.1.1 BNN-NCP con dos entradas y una salida

Al inicio, se comenzó ampliando a dos dimensiones de entrada y una de salida, como siempre con su incertidumbre epistémica (debida al modelo) y aleatoria (debida a los datos de entrada). De manera que la arquitectura de la red corresponde con la figura 3.1 con $L = 2$ y $K = 1$. En este caso se configuró la red neuronal con 200 neuronas en cada capa .

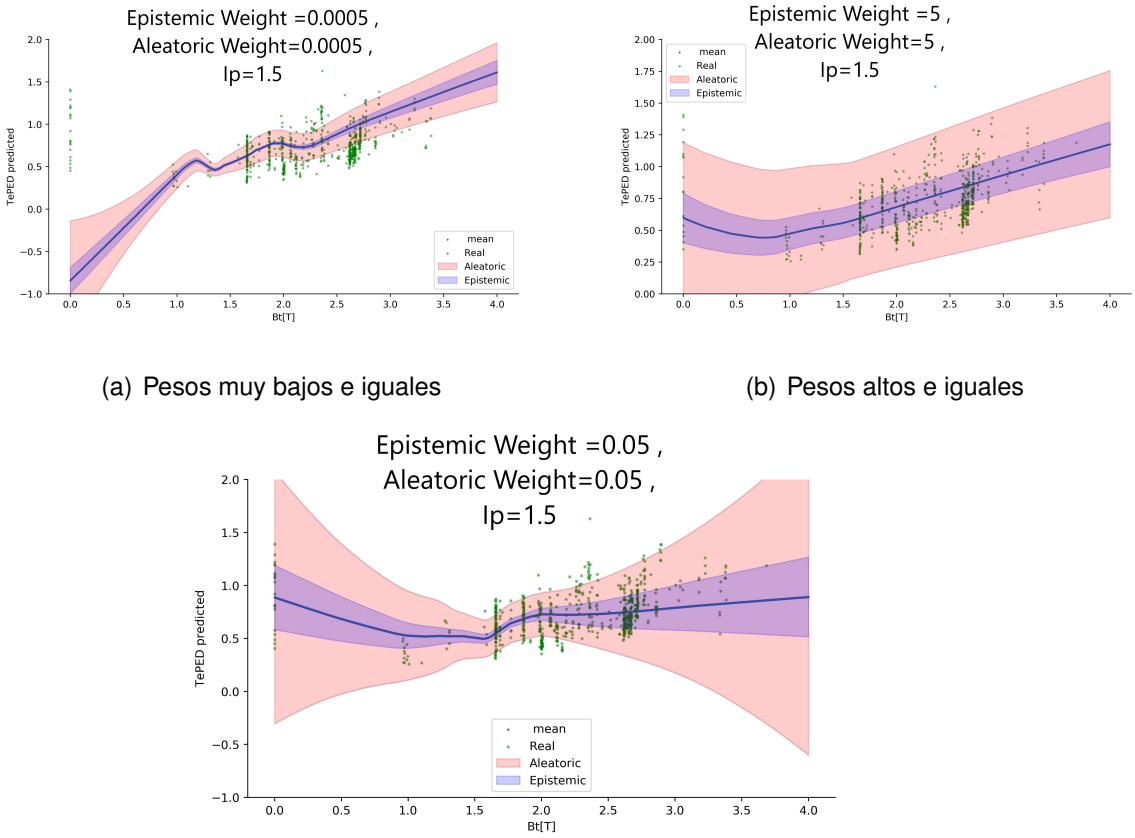
Recordemos que, en la primera distribución de salida, correspondiente a la media, obtenemos una normal unidimensional donde la media corresponde con la predicción y la desviación estándar con el error epistémico o error debido al modelo. En la segunda obtenemos una distribución normal unidimensional donde la desviación estándar corresponde con el error aleatorio o error debido a los datos y su ruido subyacente.

Esta primera aproximación nos servirá para validar el comportamiento de la BNN-NCP en un modelo multidimensional y ver si extiende los resultados de [60] en una dimensión.

En este caso, las dos variables de entrada elegidas para probar la red neuronal con los datos provenientes de JET-ILW han sido I_p y B_t, y la variable de salida Te_ped_EPED. De este modo nuestra red neuronal bayesiana con contraste de ruido a priori (BNN-NCP), estaría tratando de imitar la predicción para el pedestal de temperatura del modelo EPED con tan solo dos variables de entrada. Lo cual no esperamos que sea posible pero por ser estas dos variables de entrada tan importantes en el confinamiento del plasma esperamos que al menos arroje resultados interesantes.

Mientras se realizaban los primeros entrenamientos de la red, se hizo evidente que el rendimiento de la misma era muy sensible a algunos de los hiper-parámetros, entre los cuales destacaban los pesos asignados a las diferentes contribuciones de la función de coste. En este caso la contribución del error epistémico y la contribución del error aleatorio eran las únicas con pesos asignados manualmente. Dado que estamos en un modelo bidimensional y por tanto podemos visualizar el rendimiento de la red de manera sencilla, decidí probar con diferentes valores ya que cuando la dimensionalidad aumente en modelos posteriores esta no será una tarea tan sencilla.

De este modo en las figuras 4.1(a), 4.1(b) y 4.1(c) encontramos la salida de la red representada junto a los datos originales de la siguiente forma: se ha elegido un valor fijo para la variable de entrada Ip de 1.5 MA, y se ha comparado la predicción de la red para variable de salida $TePed$ para diferentes valores del campo magnético toroidal B en Teslas. Entonces tenemos una representación de la salida de la red con sus incertidumbres aleatorias y epistémicas junto con los datos que nos han servido de entrenamiento. Es claro que los datos de entrenamiento toman valores tanto en Ip como en Bt por lo que se parte de que esta representación es incompleta, es decir, para los datos de entrenamiento se representa la proyección de los mismos en el plano $Ip = 1.5\ MA$, así que no esperamos que las incertidumbres se ajusten fielmente, pero nos ayuda a hacernos una idea.



(a) Pesos muy bajos e iguales

(b) Pesos altos e iguales

(c) Pesos bajos e iguales

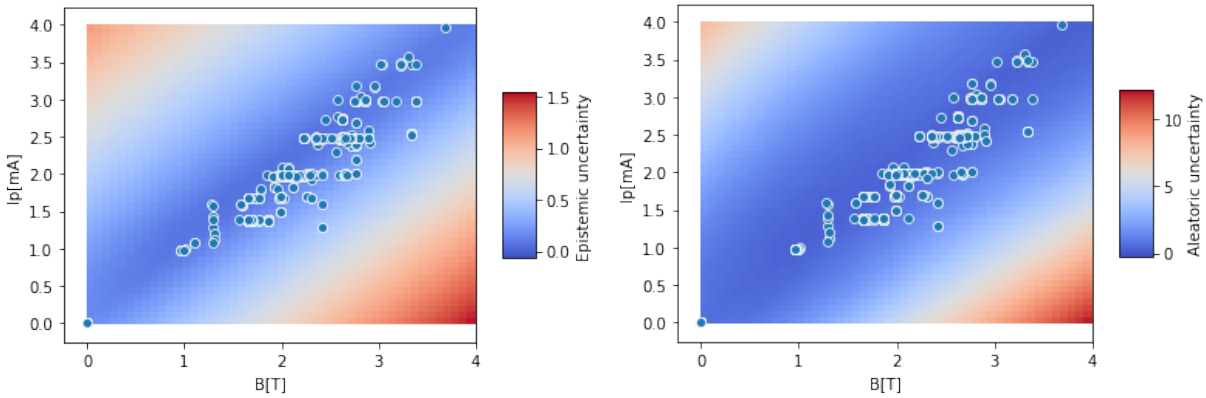
Figura 4.1: Representación de la predicción de Teped en el modelo bidimensional, tomando valor fijo $Ip = 1.5$ MA y comparando para los diferentes valores de Bt. La predicción se representa por una línea azul, el error epistémico por una franja morada, el aleatorio por una franja rosa y la proyección de los datos de entrenamiento en el plano $Ip = 1.5$ con puntos verdes. Se distinguen tres figuras según los pesos de las incertidumbres en el entrenamiento.

Recordemos que el comportamiento esperado es que donde haya datos de los que tomar referencia (datos de entrenamiento) la incertidumbre epistémica debería ser menor, pues el modelo tendría mayor fiabilidad en las zonas donde más datos existen, y donde no hay datos, el modelo extrapola por lo cual la incertidumbre del mismo debe aumentar. En cuanto al error aleatorio debería tener unos valores razonables donde se encuentra el núcleo de los datos y, o bien disminuir o bien aumentar donde no hay apenas datos de entrenamiento. Ambas afirmaciones se cumplen en el caso de la figura 4.1(c), ya que tanto el error epistémico como el aleatorio son menores en torno a los datos de entrenamiento y aumentan considerablemente según se alejan del centro de los datos. Por tanto es fácil

ver que el modelo entrenado con los pesos del orden de 0.05 y ambos iguales reproduce los resultados más deseables, si lo comparamos con las figuras 4.1(a) y 4.1(b). En la figura 4.1(a), aunque el error disminuye en las zonas donde no hay datos, este es demasiado pequeño y la extrapolación en zonas de valor del campo magnético bajo produce resultados sin coherencia (temperatura negativa). En la figura 4.1(b) apenas se aprecia ningún cambio entre las incertidumbres según la distancia con los datos, por lo tanto el resultado no es el deseado. En las tres figuras, la predicción en la zona donde hay datos de entrenamiento es coherente y se ajusta a los mismos, lo cual es una buena señal para desarrollar el modelo final. Los casos en los que los pesos para los errores epistémico y aleatorio eran diferentes arrojaban resultados menos deseables y no aportaban información extra, por lo cual decidí no mostrarlos en esta memoria.

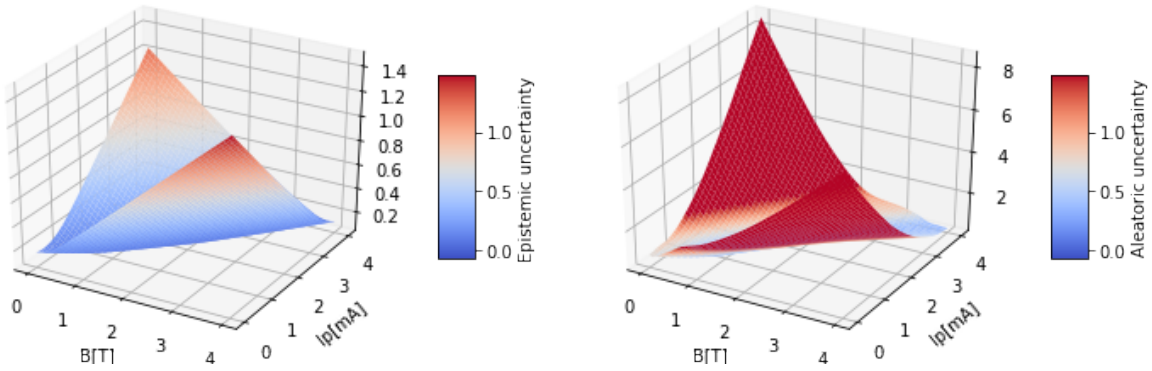
Tras habernos cerciorado de cuales son los pesos que mejor nos funcionan queremos ver la distribución del error de una manera más correcta para asegurarnos de que el comportamiento del modelo coincide con lo esperado. En las figuras 4.2(a) y 4.2(b) tenemos una representación bidimensional con mapa de colores que nos indica en que zonas del plano Ip-Bt el error epistémico y aleatorio respectivamente, aumentan o disminuyen. También se encuentran representados en este diagrama los datos de entrenamiento que nos permiten comprobar cómo el comportamiento esperado se cumple, y efectivamente en las zonas donde se encuentran los datos tanto el error epistémico propio del modelo como el error aleatorio disminuyen notablemente, dejando entrever que la fiabilidad del modelo es mayor en esta zona, mientras que en las otras está extrapolando.
En las figuras 4.3(a) y 4.3(b) tenemos la misma representación pero en este caso tridimensional para una mejor visualización de los valores de estos errores, donde vemos que el aleatorio es claramente superior al epistémico.

Tras haber comprobado que el modelo bidimensional, con la intensidad de corriente poloidal Ip y el campo magnético toroidal Bt como variables de entrada, funciona de manera deseable y ofrece valores esperados en la distribución del error para la variable de salida $Teped$ (keV), estamos en disposición de empezar a construir modelos multidimensionales que sirvan realmente como modelos sustitutos para pedestales de plasma.



(a) Error epistémico (b) Error aleatorio

Figura 4.2: Representación 2D de los errores de la predicción de Teped, epistémico y aleatorio respectivamente, en el modelo bidimensional mediante mapa de colores. A la vez se muestran los datos de entrenamiento utilizados representados por puntos azules.



(a) Error epistémico. (b) Error aleatorio.

Figura 4.3: Representación 3D de los errores, epistémico y aleatorio respectivamente, de la predicción de Teped en el modelo bidimensional, para un rango de valores de Ip y de Bt.

Proseguiremos haciendo uso de nuestro *dataset* proveniente del experimento JET-ILW.

4.2 Implementación para Modelos Sustitutos

Tras verificar que los resultados son positivos en el modelo bidimensional, llega el momento de aventurarse hacia un mayor número de dimensiones, tanto de entrada como de salida. Ya hemos visto en la sección 3.1 cómo esto no supone un inconveniente desde el punto de vista del concepto, sin embargo llevarlo a cabo puede acarrear ciertos problemas.

No debemos olvidar que el verdadero objetivo de esta investigación es elaborar modelos sustitutos que permitan acelerar los modelos de pedestales de plasma descritos anteriormente. Para ello se comienza con el modelo EPED descrito con detalle en el apartado 2.2.1.

4.2.1 Modelo sustituto de EPED

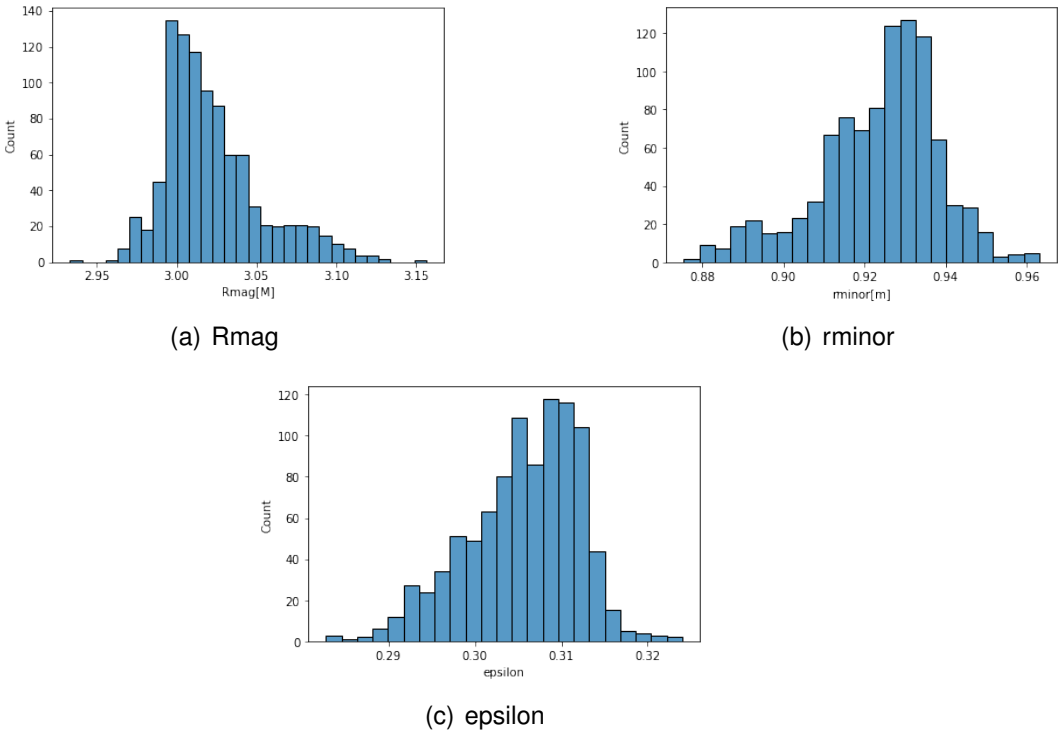
En el caso de EPED, existen 8 variables de entrada y 2 de salida. Dentro de nuestro conjunto de datos proveniente del tokamak JET, los datos de entrada se corresponden con las columnas: $Ip[MA]$, $Bt[T]$, $Triang$, $Rmag[M]$, $rminor[m]$, $kappa$, $beta\_n\_exp$ y ne_ped_exp. Y las columnas de salida son: Te_ped_EPED y $Delta\_EPED$. Estos últimos dos son los valores de salida del modelo EPED, por lo cual serán esenciales para replicar el comportamiento de este modelo.

Sin embargo, una meta de este proyecto también es reducir el número de variables necesarias para que el modelo funcione, además de adimensionalizar las variables tanto como sea posible , y durante el transcurso de la investigación, se observó que sustituyendo las variables de entrada $Rmag[M]$ y $rminor[m]$ por una variable adimensional que hemos denominado $epsilon$ se obtenían resultados equivalentes. La nueva variable, definida como $epsilon = rminor[m]/Rmag[M]$, ofrecía unos resultados satisfactorios al ser utilizada en el proceso de entrenamiento, lo cual supuso cambiar de un modelo de 8 dimensiones de entrada a uno de 7, con la pertinente mejora en el tiempo de computación.

Otro valor añadido para introducir la variable $epsilon$ es que la distribución de los datos es más ancha (su desviación estándar es mayor) que en el caso de las variables $Rmag[M]$ y $rminor[m]$, evitando de este modo que el modelo otorgue demasiada importancia a valores que tienen poca variación. De este modo en las figuras 4.4(a), 4.4(b) y 4.4(c) podemos ver como en el caso de $Rmag[M]$ la variación de los datos es del $0.1/3 \approx 3\,\%$ y para $rminor[m]$ del $0.03/0.93 \approx 3\,\%$, mientras que para el caso de $epsilon$ es de alrededor del $0.02/0.31 \approx 6\,\%$. Este aumento en la variación afecta positivamente a la importancia

que la red atribuye a cada variable.

Además del ensanchamiento de la distribución, introducir variables adimensionales como *epsilon* nos permite aplicar nuestro modelo en dispositivos diferentes a JET. Esto es debido a que el rango de valores de las variables *Rmag* y *rminor* es propio del tokamak JET, pero su relación, *epsilon*, comprende un rango de valores común a otros dispositivos, tales como *ASDEX Upgrade*, localizado en el Instituto Max Planck.



(a) Rmag (b) rminor

(c) epsilon

Figura 4.4: Histograma de los datos correspondientes a las variables Rmag[M], rminor[m] y epsilon.

Para evaluar los resultados de esta implementación, usaremos la gráfica $\beta_{p,ped} - \Delta$, tal y como se definió en el apartado 2.2.1, concretamente, la podemos observar en la figura 2.9, donde el eje horizontal corresponde con la raíz cuadrada de $\beta_{p,ped}$, por lo cual el ajuste es una linea recta. Tal y como se detalló en dicho apartado, uno de las principales características del modelo EPED se basaba en esta relación, entre el ancho del pedestal del plasma Δ y el parámetro $\beta_{p,ped}$ que corresponde con la beta plasmática poloidal. Esta beta poloidal fue definida en la ecuación 2.6. No debemos confundir la variable β_p o beta

poloidal (única para cada superficie de flujo en cada descarga), que deberá ser calculada mediante una función específica, con las variables *beta_n_exp* y *beta_n_pred*, las cuales representan el valor global de beta. La beta poloidal en el pedestal será referida como β_p o $\beta_{p,ped}$ indistintamente, sin embargo β_n es un valor global para cada descarga del plasma.

Tras implementar el método BNN-NCP en el modelo EPED, hay varios hiper-parámetros que se pueden modificar, lo que incrementa la complejidad a la hora de seleccionar la configuración adecuada del aprendizaje. Entre ellos se encuentran los coeficientes del término KL del error aleatorio y epistémico, un coeficiente para el término NLL, el número de *epochs* o ciclos de entrenamiento (iteraciones sobre los datos en el proceso de entrenamiento), el número de neuronas en cada capa y la desviación de cada variable de entrada y de salida para generar los datos OOD. Los cambios en estos hiper-parámetros conllevan variaciones importantes en el resultado, por ello, una gran parte de la investigación se ha centrado en explorar los rangos de valores donde el rendimiento del modelo es mejor.

El proceso de entrenamiento es por lo tanto una tarea delicada que consume la mayor parte del tiempo en el desarrollo de la investigación. En el caso de EPED los parámetros que mejor han funcionado, y que son dignos de ser comentados son: coeficiente KL epistémico: 0.01, en el caso del coeficiente KL aleatorio: 20, coeficiente del término NLL para Te_{ped}:1, coeficiente del término NLL para $Delta_{ped}$: 5, número de *epochs*: 1000, numero de neuronas en cada capa: 15 y 7 respectivamente. La diferencia en los coeficientes para la divergencia KL se debe a que debemos ajustar manualmente la manera en la que la red optimiza cada término para obtener resultados más satisfactorios, y se encontró que el término aleatorio necesitaba un peso mayor que el epistémico para que los resultados se ajustaran al comportamiento esperado. Del mismo modo ocurre en el caso del término NLL. En cuanto al número de ciclos de entrenamiento o *epochs*, 1000 resultó ser un valor adecuado teniendo en cuanta la relación entre tiempo de computación y resultados. Al inicio del proyecto el número de neuronas, al igual que en las secciones 2.5.1 y 4.1.1, era de 200 en cada capa, sin embargo, los resultados son similares si bajamos este número drásticamente, lo que mejora el rendimiento a nivel computacional del modelo. En la

imagen 4.5 tenemos la representación de diferentes parámetros que nos indican el rendimiento en el proceso de entrenamiento de la red neuronal. Vemos que el mayor descenso de estos parámetros se produce en las fases más tempranas de entrenamiento, después de esto el descenso es más suave, o como ocurre en el caso de los términos KL *EPI Loss* y *Alea Loss* (epistémico y aleatorio respectivamente) incluso llega a permanecer constante. Un aspecto típico en las funciones de perdida con términos NLL y KL es que debemos de encontrar el equilibrio entre minimizar uno u otro, ya que es común que cuando solo se minimiza uno de los términos (los KL por ejemplo), el otro tiende a diverger hacia valores dispares. A pesar de que la convergencia del término NLL no es tan evidente, puede observarse mediante la métrica del error cuadrático medio (MSE) para ambas variables de salida que el aprendizaje se está llevando a cabo de manera satisfactoria.

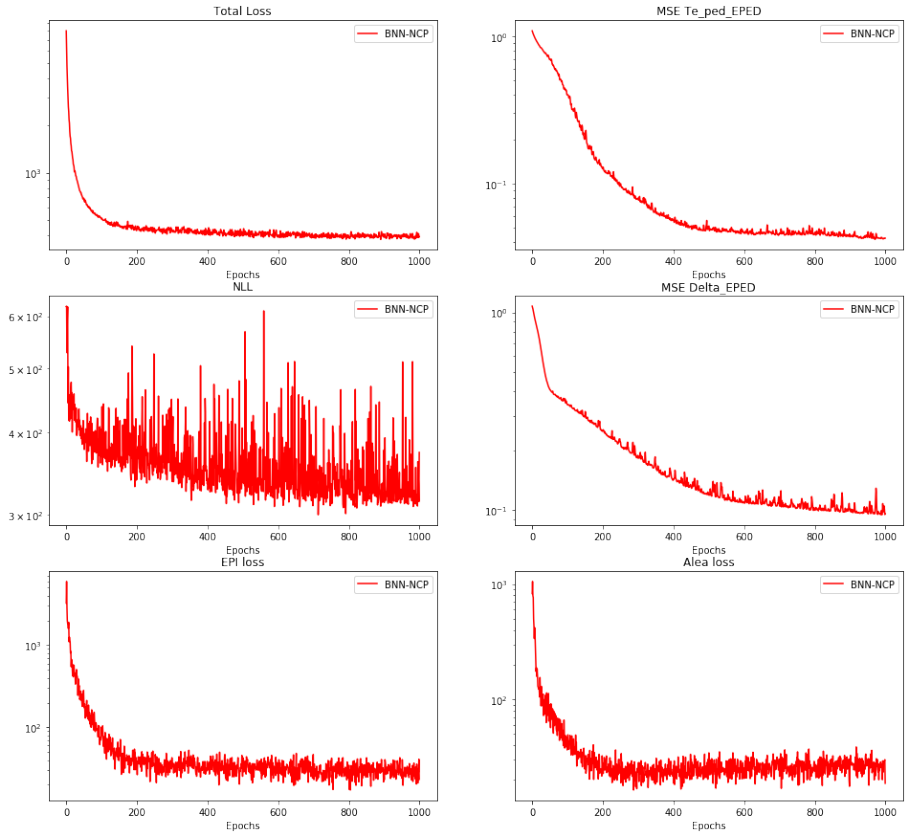


Figura 4.5: Gráficas del valor de las diferentes métricas utilizadas y componentes de la función de coste en función del ciclo de entrenamiento o epoch. En la primera columna: Valor total de la función de coste, término NLL y término de divergencia KL del error epistémico. En la segunda columna: Error cuadrático medio (MSE) respecto de los datos de entrenamiento de las variable de salida $Te_{ped,EPED}$ y $Delta_{EPED}$ respectivamente, y término de divergencia KL del error aleatorio.

Por otro lado, la manera más adecuada de medir el rendimiento y la conveniencia del modelo es el coeficiente c de nuestro ajuste en la gráfica $\beta_p - \Delta$ y la similitud con la gráfica de la figura 4.6. En el experimento [25] se obtiene un coeficiente c_1 de 0.076 en la ecuación $\Delta = c_1\sqrt{\beta_{p,ped}}$, sin embargo sabemos que esto puede variar entre diferentes tokamak e incluso entre diferentes experimentos, siempre con valores próximos. En el caso del modelo EPED para nuestros datos, originalmente se obtiene el coeficiente $c_{EPED} = 0.061$ y este será el coeficiente que trataremos de alcanzar para replicar el modelo. Se puede observar esta relación en los datos JET en la figura 4.6, donde se muestra en rojo la curva de ajuste para la raíz cuadrada. Como vemos el ajuste es bastante fiel a los datos, y podemos confirmar que esta relación se cumple, cosa que no debería sorprendernos pues el modelo EPED lleva consigo esta restricción.

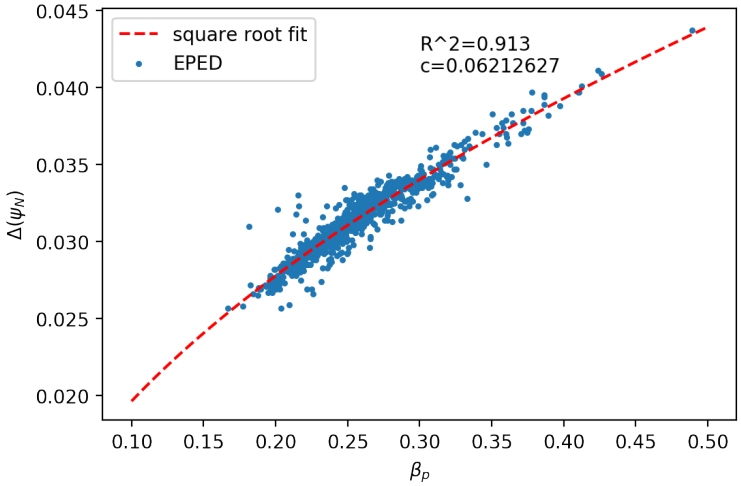
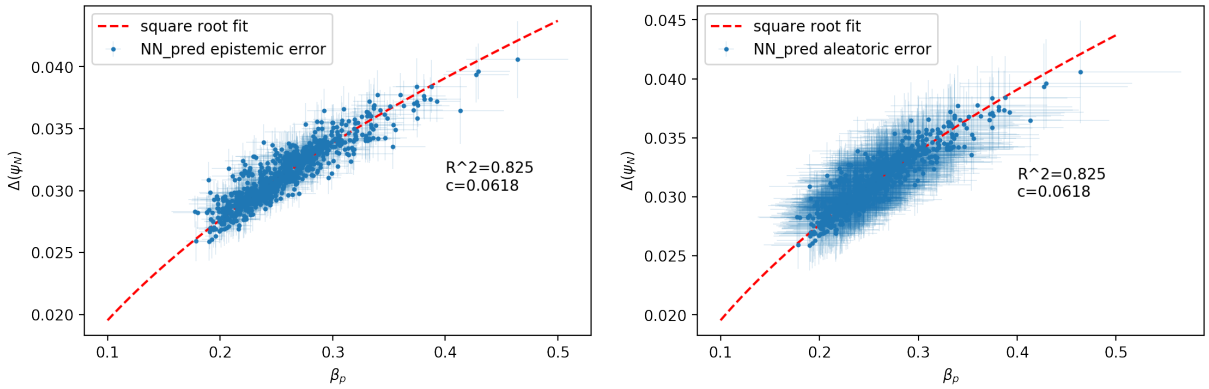


Figura 4.6: Relación entre la anchura del pedestal Delta y el parámetro del plasma beta poloidal, para el modelo EPED en el conjunto de datos obtenidos del tokamak JET.

En las figuras 4.7(a) y 4.7(b) encontramos la predicción de la red neuronal para la totalidad de los datos, es decir, tanto el conjunto de datos de entrenamiento (90 % del *dataset*) como el test. Lo encontramos representado en una gráfica Δ vs $\beta_{p,ped}$, en el caso de la figura 4.7(a) con su error epistémico o debido al modelo, y en la figura 4.7(b) con su error aleatorio o debido al ruido subyacente de los datos. Recordemos que este es un modelo con 7 dimensiones de entrada, luego no tenemos posibilidad de realizar representaciones gráficas donde se nos ofrezca información de todas las variables y de la forma en la que trabaja el modelo en ellas. Sabemos que la propia Δ_{EPED} es una de las salidas del



(a) Error epistémico.

(b) Error aleatorio.

Figura 4.7: Representación de la predicción del método BNN-NCP para el modelo EPED, en los datos JET, junto con sus errores epistémico y aleatorio respectivamente, en gráfica Δ *vs* $\beta_{p,ped}$.

modelo, sin embargo, para calcular $\beta_{p,ped}$ también usamos la otra salida, $Te_{ped,EPED}$, con la cual $\beta_{p,ped}$ mantiene una relación de proporcionalidad directa (ver ecuación 2.6, componente poloidal de β). Teniendo en cuenta la relación entre la temperatura y presión, la formula detallada para calcular $\beta_{p,ped}$ se encuentra en el código (Anexo 5.3).
Es por esto que la gráfica Δ vs $\beta_{p,ped}$ es un excelente indicador de la calidad con la que estamos elaborando el modelo.

Es claro como, al igual que en el modelo bidimensional descrito en el apartado 4.1.1, la incertidumbre aleatoria supera con creces a la incertidumbre epistémica. Esto nos da una idea de que el error de nuestro modelo, que es apreciablemente bajo si lo comparamos con la propia predicción, también lo es comparado con el error proveniente del ruido subyacente de los datos.

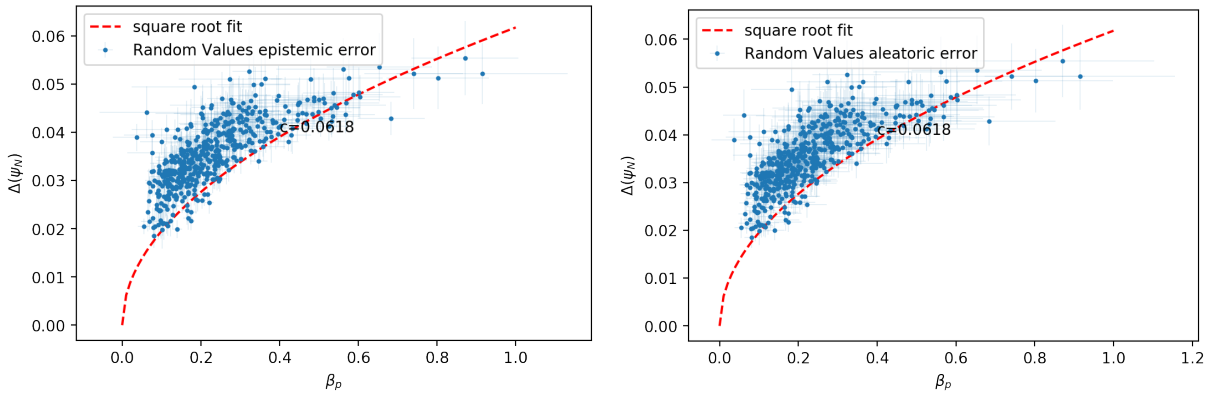
Sin embargo, la principal característica en la que estamos interesados en las figuras 4.7(a) y 4.7(b) es la constante c y el ajuste a la función raíz cuadrada. Como es evidente, no solo esta constante es muy similar a la de la figura 4.6, si no que los datos también se ajustan razonablemente bien a la curva roja al igual que en la figura 4.6, como podemos ver mediante el índice $R^2 = 0.825$.

Ahora bien, existe todavía un aspecto muy importante que esta representación no cu-

bre totalmente. Se trata del error epistémico y aleatorio en las regiones OOD, o lo que es lo mismo, las regiones alejadas de los datos. Dado que las figuras 4.7(a) y 4.7(b) tan solo nos ofrecen las predicciones para los datos de entrada provenientes del conjunto de datos totales, descrito en el apartado 3.3, entonces no tienen capacidad para mostrarnos ningún tipo de información sobre las regiones OOD. Este tampoco es un problema sencillo, de nuevo debido a la multi-dimensionalidad a la que nos enfrentamos. Con 7 dimensiones de entrada ya no podemos realizar gráficas en 2D como las de las figuras 4.2(a) y 4.2(b) pertenecientes al modelo bidimensional, y las gráficas en las que dejamos el resto de dimensiones con un valor fijo y observamos cómo varía el error con respecto a una variable carecen de sentido con tal número de dimensiones, a diferencia del caso bidimensional con las figuras 4.1(a), 4.1(c) y 4.1(b).

La solución en este caso pasa por generar datos aleatorios de entrada y introducirlos en nuestro modelo para representar los valores de salida mediante la gráfica Δ vs $\beta_{p,ped}$, junto con sus respectivas incertidumbres. De esta manera podemos observar al menos dos cosas: cómo se comporta el modelo con datos artificialmente generados, lo que puede ofrecernos información adicional sobre el mismo, y si el modelo sigue el comportamiento esperado en las zonas OOD y en la zona de los datos. Los datos aleatorios se generan a través de una distribución uniforme para cada una de las variables de entrada, cuyo rango ha sido ampliado para alcanzar regiones OOD. Se muestrean los datos desde la mitad del mínimo de la columna correspondiente a cada variable, hasta 1.5 veces el máximo de la misma columna. De este modo conseguimos un rango más amplio sobre el que hacer nuestras predicciones.

Por consiguiente, en las figuras 4.8(a) y 4.8(b) observamos la predicción obtenida para los datos aleatorios que hemos generado, junto con su error epistémico y aleatorio respectivamente. Además se presenta la misma curva de ajuste que para la predicción original de los datos JET. De este modo nos podemos hacer una idea de dónde se encuentra el conjunto de datos original. A primera vista, llama la atención cómo las predicciones, a pesar de provenir de datos generados de manera aleatoria, se ciñen razonablemente bien a la curva de ajuste y a la zona de los datos. Sin embargo, a medida que la curva asciende



(a) Error epistémico.

(b) Error aleatorio.

Figura 4.8: Representación de la predicción del método BNN-NCP aplicado al modelo EPED, para datos aleatoriamente generados, junto con sus errores epistémico y aleatorio respectivamente, en gráfica Δ *vs* $\beta_{p,ped}$.

estas predicciones se esparcen más por el espacio, concretamente hacia valores mayores de Beta. En primer lugar, el hecho de que las predicciones se ciñan tanto a la curva para datos aleatorios podría interpretarse como que la generación de los datos no ha sido suficientemente amplia. Sin embargo esto no es cierto: la realidad es que el modelo está acostumbrado a trabajar en esa zona y el hecho de que las predicciones las haga en esa región (con un cierto margen) es un indicativo de que el modelo ha aprendido correctamente. De hecho, en las zonas en las que hay menos datos de entrenamiento, como por ejemplo en $\beta_p > 0.4$, la dispersión de las predicciones es mayor que en el resto. Esto nos dice que el modelo está actuando correctamente. Además no debemos olvidar que la relación $\Delta_{ped} = c\sqrt{\beta_{p,ped}}$ es una restricción que se cumple en el modelo EPED, luego que nuestro modelo trabaje en esa zona no nos debería resultar extraño.

El otro aspecto que buscábamos de esta representación es conocer la distribución del error en el plano $\Delta - \beta_p$. No obstante, existe otro modo de visualizar el comportamiento del error, que es representar la incertidumbre aleatoria o epistémica, con respecto a la distancia de la predicción al centroide de los datos. Es decir, relacionar la distancia con respecto a la región de los datos de una predicción, con su incertidumbre, para efectivamente comprobar si esta crece o decrece con la distancia, en otras palabras, descubrir que ocurre en las zonas OOD.

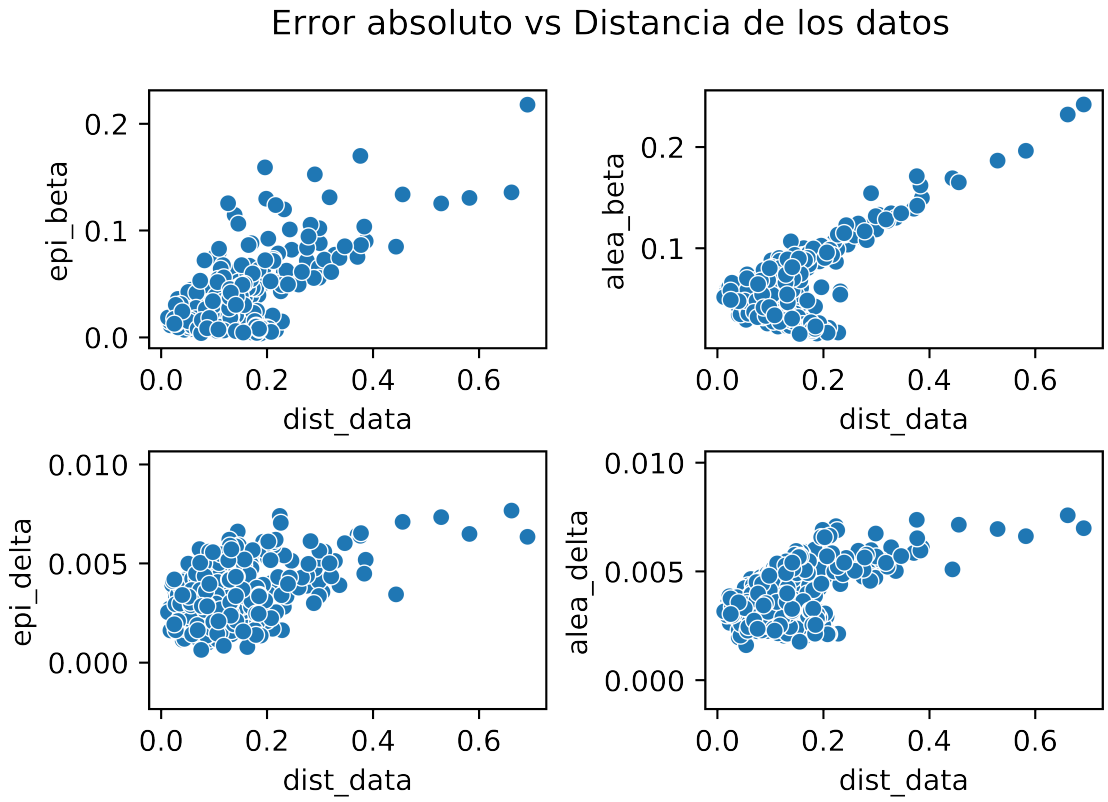


Figura 4.9: Incertidumbre absoluta epistémica y aleatoria de las predicciones del modelo sustituto de EPED, para las variables β_p y Δ, en función de la distancia con respecto al centroide de los datos de entrenamiento.

En la figura 4.9 observamos las 4 incertidumbres posibles que se extraen de la gráfica $\beta_p - \Delta$, en función de la distancia de la predicción al centroide de los datos de entrenamiento. Estas incertidumbres son las epistémica y aleatorias de las variables β_p y Δ. A simple vista podemos deducir que cada una de las incertidumbres crece con la distancia al centroide de los datos, lo cual no es otra cosa que el comportamiento que buscamos en el modelo. Del mismo modo se aprecia como los puntos cercanos al centroide tienen las incertidumbres más bajas, es decir, en estas zonas el modelo es fiable, sin embargo, en las zonas más alejadas (OOD) la incertidumbre es mayor por lo que el modelo está extrapolando y las predicciones no son tan certeras. Por último, podemos diferenciar cómo en las 4 gráficas, pero sobre todo en la correspondiente a la incertidumbre aleatoria de beta, hay una pequeña desviación de la tónica general que tiene tendencia decreciente, se trata de las predicciones cercanas al 0 de una variable (en especial para β_p) cuya incerti-

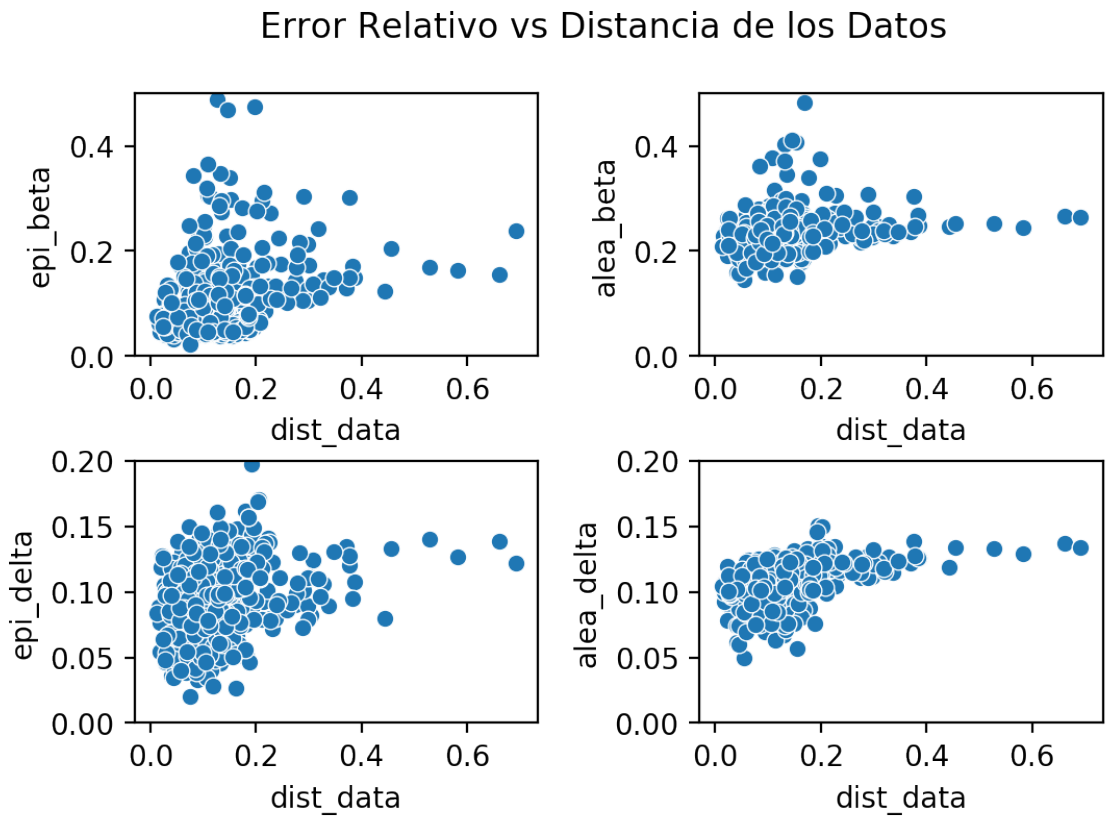


Figura 4.10: Incertidumbre relativa epistémica y aleatoria de las predicciones del modelo sustituto de EPED, para las variables β_p y Δ, en función de la distancia con respecto al centroide de los datos de entrenamiento.

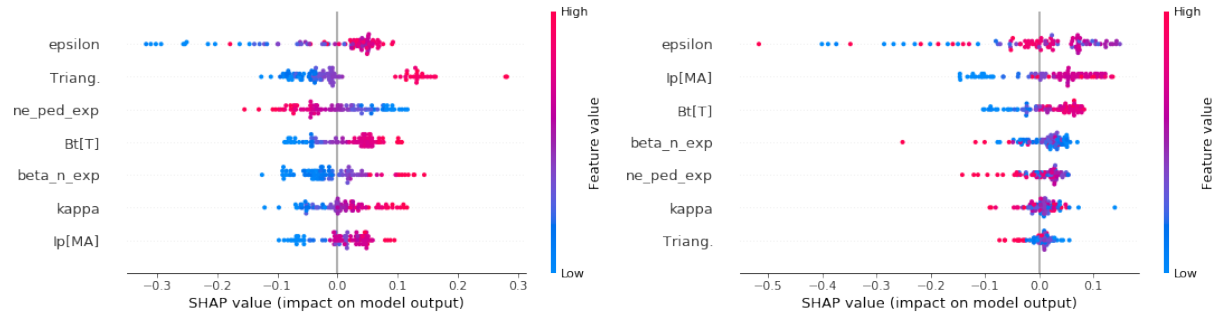
dumbre no aumenta por tratarse de valores pequeños de la variable. Es decir, disminuye su incertidumbre absoluta pero no la relativa. Lo podemos ver en la figura 4.10, donde se presentan las incertidumbres relativas. A pesar de que en esta figura el crecimiento al alejarse de los datos es menos marcado, todavía se puede percibir con claridad. Notar que las incertidumbres aleatorias tienden a ser mayores que las epistémicas, es una característica que podremos observar en todo el proyecto.

Por ende, comprobamos como nuestro modelo posee el comportamiento deseado. Además replica fielmente los resultados del modelo EPED, con todas las ventajas que esto supone a la hora de obtener información del mismo de una manera mucho más ágil y versátil. Es por ello que durante el transcurso de la investigación nos decidimos a ir un paso más adelante y tratar de conocer la importancia de cada variable dentro del modelo EPED.

Aunque se aleje del objetivo estricto del proyecto, medir la importancia de cada variable en el modelo puede ser relevante a fin de observar si se están utilizando relaciones conocidas entre las variables, y también con el fin de observar otras nuevas, de este modo podemos llegar a destapar la física oculta tras el comportamiento del plasma.

Asimismo, para confeccionar una primera aproximación nos vamos a limitar a utilizar los valores SHAP (*SHapley Additive exPlanations*) [63]. Estos valores nos ofrecen para cada salida del modelo, la contribución de cada variable de entrada para el valor final. Por tanto también es capaz de categorizar la importancia y el impacto de cada variable en el resultado final. Existen varias formas de representar esta información de manera gráfica, pero la manera utilizada en las figuras 4.11(a) y 4.11(b) será la más conveniente. En ellas podemos ver las variables de entrada ordenadas de mayor a menor según su importancia para cada una de las dos variables de salida. Además se aprecia como los valores altos o bajos de una determinada variable de entrada afectan positivamente o negativamente (indistintamente) a la variable de salida mediante el valor SHAP. Si este valor (SHAP) es positivo, significa que ese dato de la variable de entrada está provocando que la predicción de la variable de salida aumente, asimismo si el valor SHAP es negativo, el dato provoca que la predicción disminuya.

Siguiendo este razonamiento advertimos como para algunas variables de entrada el comportamiento está muy marcado mientras que para otras es más difuso. En ambas predicciones, $epsilon$ es la variable de entrada con más relevancia para la red neuronal. Si nos fijamos en el análisis de la predicción de $Te_{ped,EPED}$ en la figura 4.11(a), vemos como en la mayoría de variables el comportamiento está muy marcado, es decir, los puntos rojos tienen un signo diferente de valor SHAP que los puntos azules. No ocurre así en la predicción de $Delta_{EPED}$ en la figura 4.11(b), donde puntos rojos y azules se entremezclan dejando ver relaciones más difusas, propias de un modelo complejo. Además, en ambas figuras el orden de importancia de las variables no es el mismo, lo que por otro lado es esperable pues las relaciones físicas subyacentes son diferentes para ambas variables.



(a) $Te_{ped,EPED}$ (b) $Delta_{EPED}$

Figura 4.11: Representación de la importancia de cada variable de entrada (de arriba a abajo, de más importante a menos) y su contribución a cada una de las variables de salida $Te_{ped,EPED}$ y $Delta_{EPED}$, para un conjunto test de 84 entradas proveniente del dataset original, y para el modelo sustituto de EPED. Calculado mediante los valores SHAP. En rojo valores más altos de la variable de entrada y en azul los más bajos. Valor SHAP positivo significa que provoca que el valor de salida aumente, negativo que disminuya.

A pesar de lo interesante de las figuras 4.11(a) y 4.11(b), y del potencial de la propia herramienta SHAP, a lo largo de la investigación hemos podido advertir ciertas variaciones significativas en la importancia de las variables y los valores SHAP para diferentes parámetros de entrenamientos del modelo. Esto significa que a pesar de que SHAP es una herramienta muy informativa, sus resultados han de ser tomados con cierta cautela pues podrían albergar cierta desconfianza.

Tras haber entrenado y optimizado la red BNN-NCP, y tras comprobar la veracidad de las predicciones, el comportamiento del nuevo modelo e incluso la relevancia de las variables, podemos decir que tenemos un modelo sustituto completamente funcional para el modelo de pedestal EPED. Esto permitirá, en futuras investigaciones, obtener la salida de este modelo sin necesidad de correr el modelo EPED, lo cual resultaría mucho más costoso computacionalmente. Además, acerca a EPED a su aplicación en tiempo real, lo cual supone un salto gigante a la hora de realizar experimentos de fusión nuclear. La inteligencia artificial comienza a ser una herramienta imprescindible en fusión nuclear.

4.2.2 Modelo sustituto de EuroPED

El siguiente paso después de desarrollar un modelo sustituto para EPED es hacer lo propio para EuroPED, un modelo más complejo que contiene al propio EPED. Tal y como se detalló en el apartado 2.2.2, el modelo EuroPED extiende su funcionalidad a diferentes tokamaks, no solo para el DIII-D como el caso de EPED, y no utiliza variables que son desconocidas antes del experimento, como puede ser la beta total o la densidad electrónica en el pedestal, variables que sí utiliza EPED. Por lo cual se trata de un modelo más refinado y por ende más complejo.

El modelo EuroPED, tal y como lo vamos a replicar, toma 9 variables de entrada y 3 de salida. Las variables de entrada corresponden con las columnas del *dataset* JET: $Ip[MA]$, $Bt[T]$, $Triang$, $Rmag[M]$, $rminor[m]$, $kappa$, $P\_tot[MW]$, $Gas[1e22/s]$ y $Zeff\_h$. Y las variables de salida con: $Delta\_SC$, Te_ped_SC y $neped\_pre\_2$. El sufijo *SC* significa auto-consistente (*self-consistent* en inglés), debido a la naturaleza auto-consistente de las predicciones de EuroPED, concepto detallado en la sección 2.2.2.

Sin embargo, al igual que en el modelo sustituto para EPED, procederemos a reemplazar las columnas $Rmag[M]$ y $rminor[m]$, por $epsilon = rminor[m]/Rmag[M]$, lo que deja un total de 8 variables de entrada. Además, durante el desarrollo del modelo, se concluyó que se podía llevar a cabo la sustitución de la variable $Ip[MA]$ por una variable adimensional que relaciona $Ip[MA]$, $Bt[T]$ y $rminor[m]$, la cual hemos bautizado como *mu* y se define: $mu = (\mu_0/(2\pi)) \cdot 10^6 \cdot Ip[MA]/(Bt[T] \cdot rminor[m])$. Tal y como se ha comentado, una de las ventajas de adimensionalizar tantas variables como sea posible es obtener un modelo que pueda ser extrapolable a datos provenientes de diferentes dispositivos tokamak. De esta forma, las 8 variables de entrada de nuestro modelo sustituto serían: *mu*, $Bt[T]$, $Triang$, $epsilon$, $kappa$, $P\_tot[MW]$, $Gas[1e22/s]$ y $Zeff\_h$.

A la hora de configurar y entrenar la red, la principal diferencia con EPED es añadir una variable de entrada y de salida. Los valores de los hiper-parámetros utilizados son los mismos, excepto para el número de ciclos de entrenamiento el cual ampliamos has-

ta 1500 *epochs*, y para los coeficientes correspondientes al término NLL en la función de coste para cada una de las variables de salida, en este caso: 2 para $Delta$, 3 para Te_{ped} y 5 para ne_{ped}. De nuevo, estos valores se han obtenido a través del método prueba y error, habiendo tomado una gran cantidad de tiempo entrenar el modelo para los diferentes valores. Esta práctica es muy común en problemas de redes neuronales, dado el conocido comportamiento de «caja negra» que alberga su funcionamiento interno. De esta forma se trata de optimizar el proceso observando el rendimiento del entrenamiento, tal y como podemos ver en la figura 4.12. En esta figura además de observar el valor de la función de coste para cada ciclo de entrenamiento, podemos ver el error cuadrático medio de las tres variables de salida, el cual está estrictamente relacionado con el término NLL. Además, también observamos los términos de la divergencia KL relacionados con el error epistémico y con el error aleatorio. De este modo, dado que en las representaciones están incluidos los coeficientes, nos es posible observar la manera en la que la red está optimizando y a qué términos está otorgando más importancia, y por tanto ajustar los coeficientes manualmente.

El número de *epochs* es un factor de gran relevancia a tener en cuenta a lo largo del entrenamiento. Si tomamos un número demasiado bajo esto puede provocar que el aprendizaje del modelo sea insuficiente y por tanto que sus predicciones no sean acertadas. En una red neuronal convencional, si tomamos un número demasiado grande de ciclos de entrenamiento el resultado suele ser un modelo sobre-entrenado, es decir, un modelo que ha iterado sobre los mismos datos tan repetidamente, que este se ajusta no solo a la tendencia de los datos si no también al ruido subyacente, lo que genera un error mucho mayor en predicciones de datos sobre los que no ha entrenado, como por ejemplo el conjunto datos *test*. Sin embargo, nuestro particular modelo no sufre estos efectos debido al efecto regularizador de la divergencia KL, la que a cambio nos ofrece un efecto no deseado: según aumentamos el número de *epochs* encontramos que la convergencia de los diferentes términos de la función de coste se debilita. Es por esto que tanto el modelo sustituto de EPED como el de EuroPED no mejoran con un mayor número de iteraciones de la red, tal y como podemos observar en la figura 4.13, donde se muestra el rendimiento de la red neuronal bayesiana a lo largo de diez mil iteraciones, y donde se puede apreciar la falta

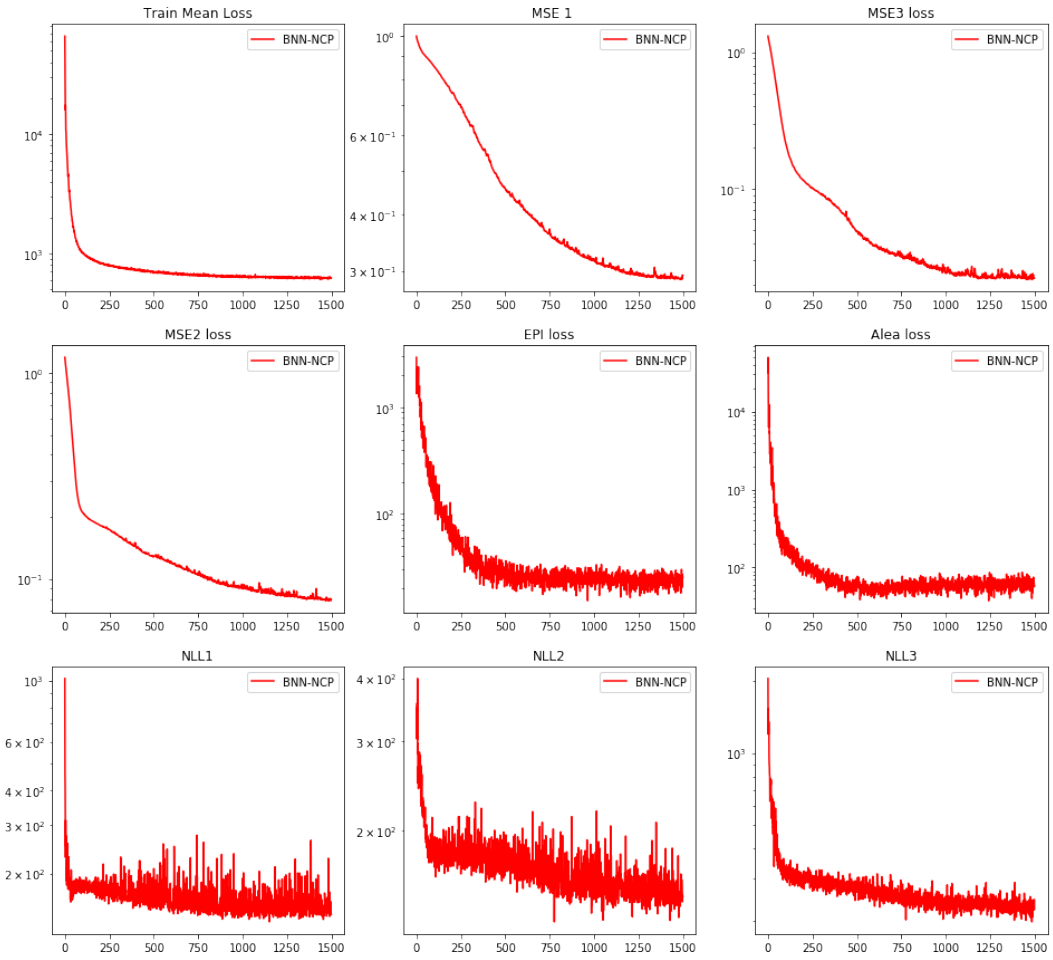


Figura 4.12: Gráficas del valor de las diferentes métricas utilizadas y componentes de la función de coste en función del ciclo de entrenamiento o epoch. En la primera fila: Valor total de la función de coste, error cuadrático medio (MSE) respecto de los datos de entrenamiento de las variables de salida $Delta_{SC}$ y $ne_{ped,pred}$. En la segunda fila: MSE para la variable de salida $Te_{ped,SC}$, término de divergencia KL del error epistémico y aleatorio. En la tercera fila: los términos NLL de las tres variables: $Delta_{SC}$, $Te_{ped,SC}$ y $ne_{ped,pred}$.

de convergencia, sobre todo a partir de las 2000 *epochs*.

Los mecanismos para medir el rendimiento de este modelo sustituto van a ser en su mayoría análogos a los utilizados en el apartado anterior. Es por ello que comenzamos mostrando la gráfica $\beta_p - \Delta$ en la figura 4.14 correspondiente a los datos de EuroPED, de los cuales tratará de aprender la red neuronal. En este caso, para calcular β_p no utilizaremos los datos experimentales de ne_{ped} alojados en la variable *ne_ped_exp*, si no que haremos uso de la predicción de EuroPED, alojada en la variable $neped\_pre\_2$. En el caso del modelo sustituto de EuroPED deberemos también calcular β_p mediante la predicción de las

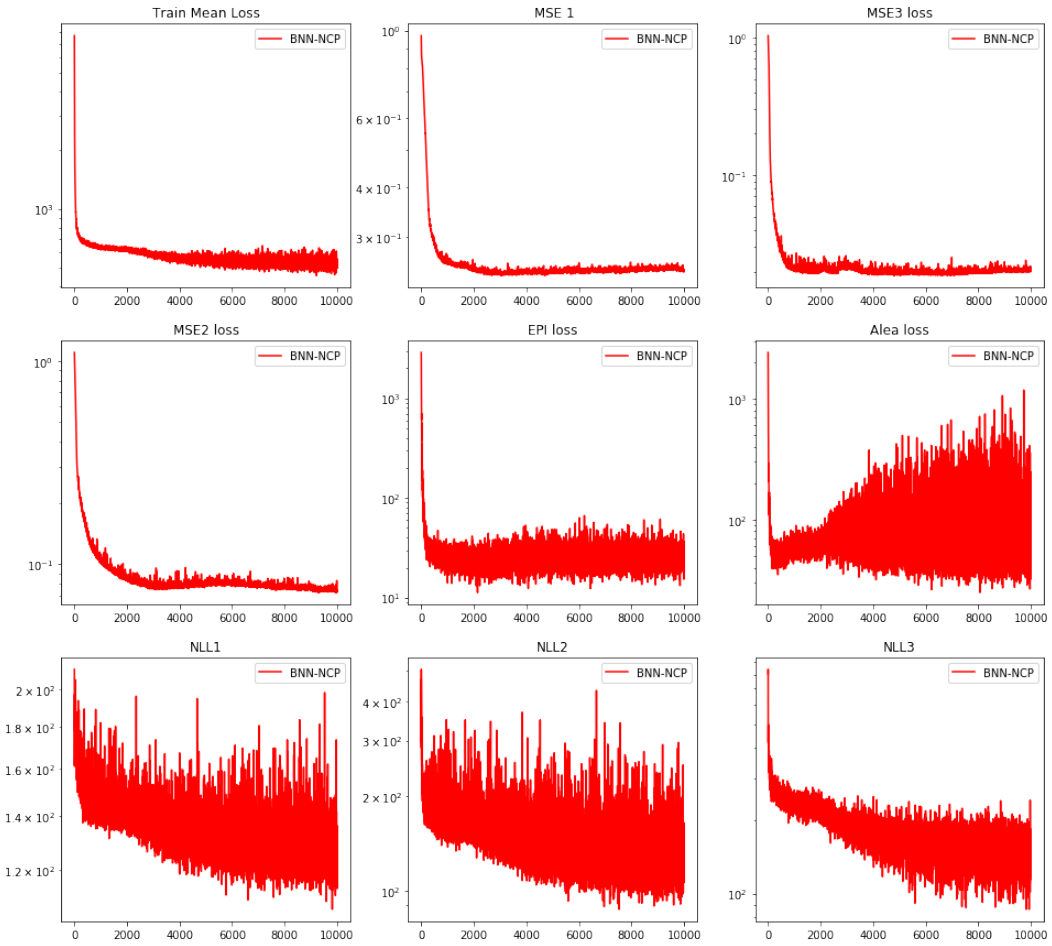


Figura 4.13: Gráficas del valor de las diferentes métricas utilizadas y componentes de la función de coste en función del ciclo de entrenamiento para 10000 epochs. En la primera fila: Valor total de la función de coste, error cuadrático medio (MSE) respecto de los datos de entrenamiento de las variables de salida $Delta_{SC}$ y $ne_{ped,pred}$. En la segunda fila: MSE para la variable de salida $Te_{ped,SC}$, término de divergencia KL del error epistémico y aleatorio. En la tercera fila: los términos NLL de las tres variables: $Delta_{SC}$, $Te_{ped,SC}$ y $ne_{ped,pred}$.

variables Te_{ped} y ne_{ped} utilizando la fórmula descrita en la ecuación 2.6. Es por esto que la representación $\beta_p - \Delta$ continua siendo un excelente mecanismo para medir el rendimiento del modelo sustituto respecto del modelo original.

Como vemos la dispersión de los datos es mayor en la figura 4.14 correspondiente a EuroPED que en la figura 4.6 correspondiente a EPED. Este no es un detalle menor, si no que es uno de los puntos clave de la investigación, la razón radica en la propia naturaleza de EuroPED y será explicada con detalle en la sección 4.3. Por otro lado, en la figura 4.14 podemos apreciar claramente dos conjuntos de datos diferenciados, esta separación

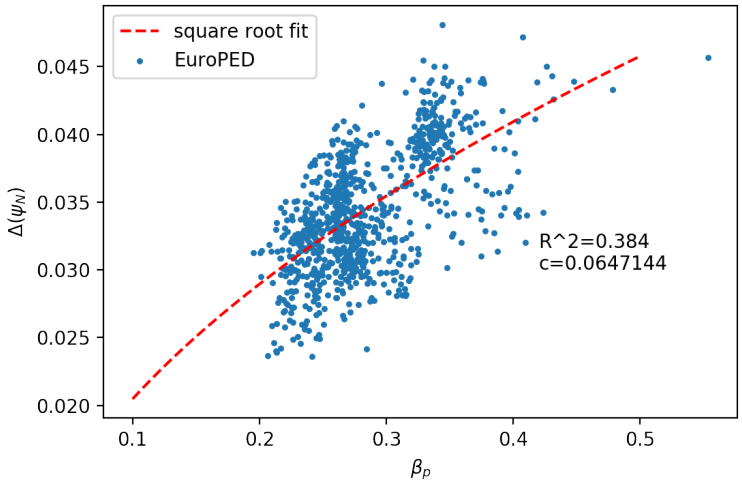
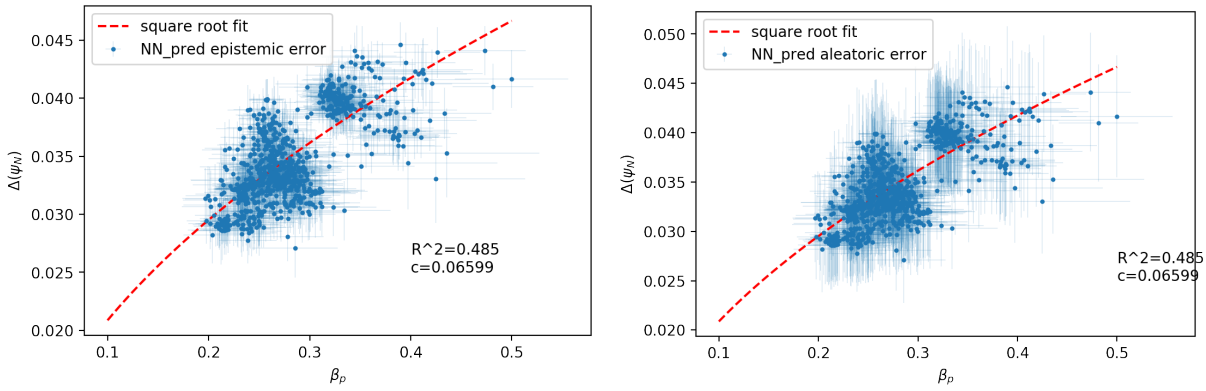


Figura 4.14: Representación de la salida del modelo EuroPED según el conjunto de datos proveniente del tokamak JET, en forma de gráfica $\Delta - \beta_p$.

se debe a la variable *Triang*. correspondiente a la triangularidad del plasma.



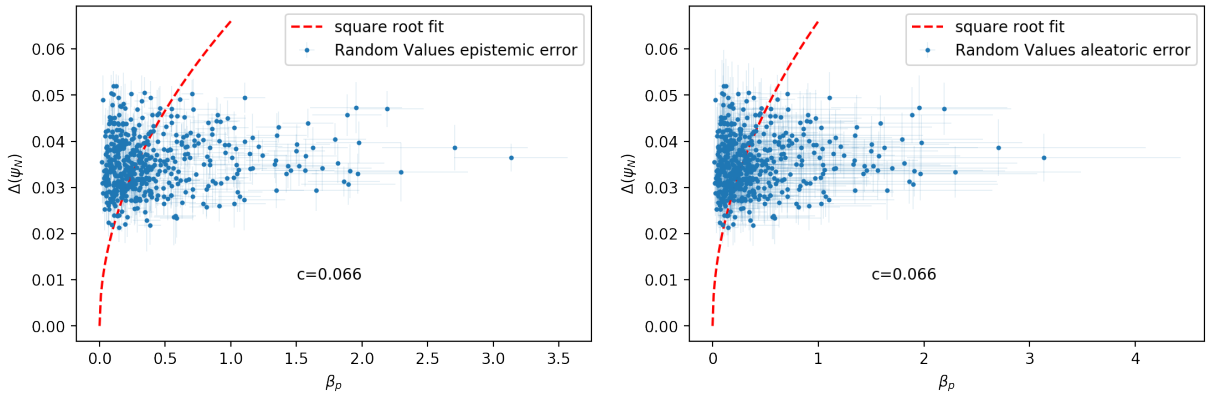
(a) Error epistémico.

(b) Error aleatorio.

Figura 4.15: Representación de la predicción del método BNN-NCP para el modelo EuroPED en los datos JET, junto con sus errores epistémico y aleatorio respectivamente, en gráfica Δ vs $\beta_{p,ped}$.

En las figuras 4.15(a) y 4.15(b) podemos observar las predicciones de la red neuronal para los datos de entrada junto con sus incertidumbres epistémica y aleatoria respectivamente, representados como siempre, en el gráfico $\Delta - \beta_p$. De aquí podemos obtener varias conclusiones. En primer lugar vemos como la distribución de los puntos de las figuras 4.15(a) y 4.15(b) es muy similar a la de los datos originales, dispuesta en la figura 4.14, esto es un resultado esperado, al igual que en el caso de EPED. En las tres figuras vemos como el ajuste de raíz cuadrada no es para nada adecuado, pero nos deja hacernos

una idea de la distribución de los datos en el espacio $\Delta - \beta_p$, además de permitirnos comparar los valores entre las gráficas de predicciones y la de datos. En particular, en la figura 4.15(a) se muestra la predicción para los datos de entrada con su error epistémico, o debido al modelo, el cual podemos apreciar que es sensiblemente mayor al mostrado en la figura 4.7(a) correspondiente al modelo sustituto de EPED, lo que puede ser debido a la dificultad que encuentra la red neuronal para capturar las relaciones internas en un modelo de mayor complejidad como es EuroPED. Asimismo, en la figura 4.15(b) encontramos las mismas predicciones pero con el error aleatorio o debido al ruido subyacente de los datos de entrada, el cual de nuevo es sensiblemente mayor que su análogo en EPED en la figura 4.7(b), esto puede de otra manera deberse a la mayor dispersión de los datos de EuroPED que es evidenciado en la figura 4.14, donde se muestra la salida deseada con una forma no muy bien definida.



(a) Error epistémico. (b) Error aleatorio.

Figura 4.16: Representación de la predicción del método BNN-NCP para datos aleatoriamente generados, junto con sus errores epistémico y aleatorio respectivamente, en gráfica Δ vs $\beta_{p,ped}$.

De nuevo, para medir el comportamiento del modelo respecto al error en las diferentes zonas, muestreamos datos de entrada aleatorios en un determinado rango y obtenemos las predicciones que nos ofrece el modelo junto con su incertidumbre. Estas predicciones son dispuestas en forma de gráfico $\Delta - \beta_p$ tal y como podemos observar en las figuras 4.16(a) y 4.16(b), donde se muestran las incertidumbres epistémica y aleatoria respectivamente. Estas figuras presentan una clara diferencia con las figuras 4.8(a) y 4.8(b) pertenecientes al modelos sustituto de EPED: la distribución de las predicciones es mucho más disper-

sa en el caso de EuroPED, y no se ciñe al ajuste de raíz cuadrada, como ocurre en EPED. Esto podría parecer un problema, pues el modelo no se ajusta tan fielmente a la zona en la que está acostumbrado a hacer las predicciones. Sin embargo no es así, es un comportamiento esperado debido a la dispersión de los datos en el propio modelo EuroPED, lo cual produce que el modelo no se ciña solo a esa región si no que también aporte cierta dispersión en la salida.

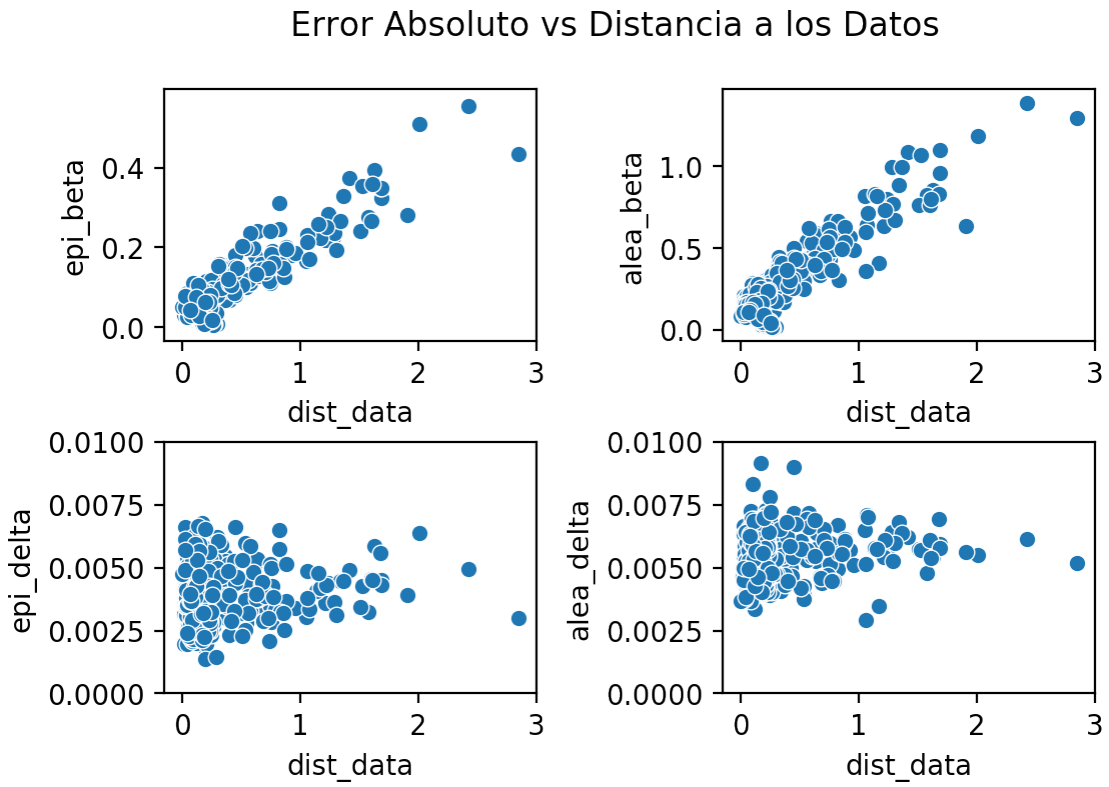


Figura 4.17: Incertidumbre absoluta epistémica y aleatoria de las predicciones del modelo sustituto de EuroPED, para las variables β_p y Δ, en función de la distancia con respecto al centroide de los datos de entrenamiento.

Para interpretar de una forma adecuada las figuras 4.16(a) y 4.16(b), debemos de hacer uso de una tercera representación, ya que al igual que ocurría en el apartado anterior, queremos conocer la distribución del error dentro y fuera de la región de los datos. Para ello nos servimos de la figura 4.17, donde se representan las incertidumbres epistémica y aleatoria de las predicciones para variables β_p y Δ de los datos aleatorios generados, con respecto a la distancia del centroide de los datos de entrenamiento. Es decir, la incertidumbre en función de la distancia a la región de los datos.

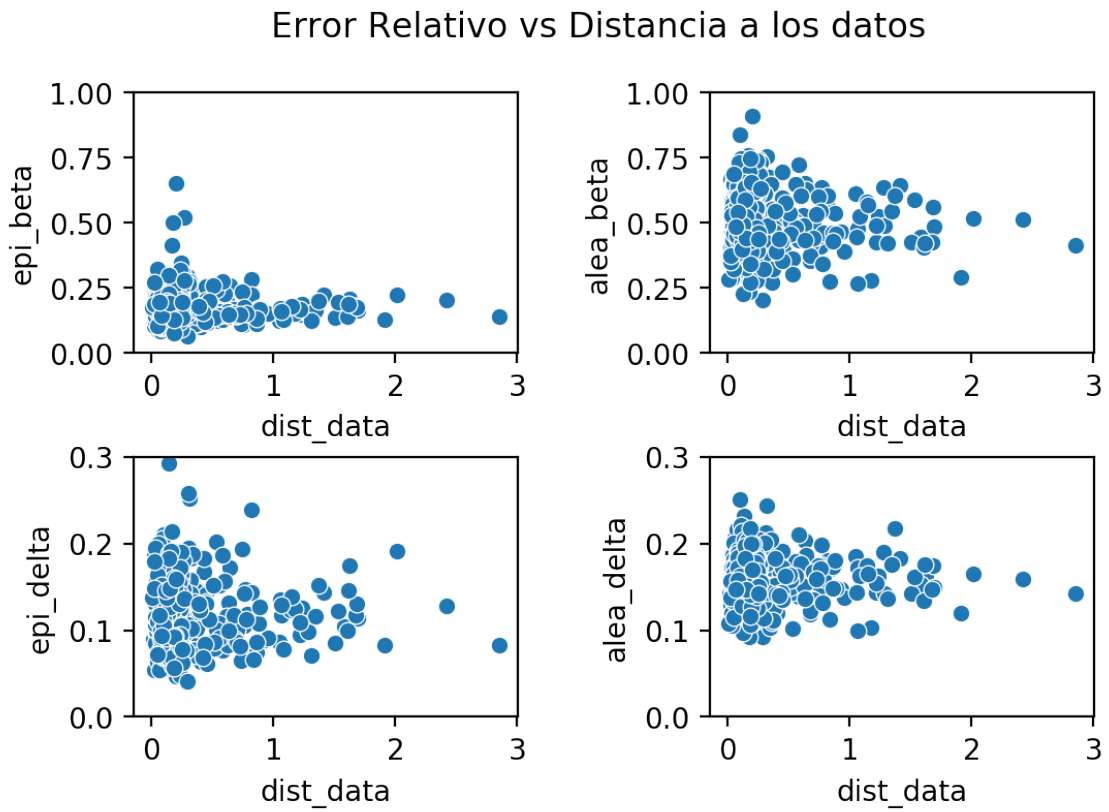


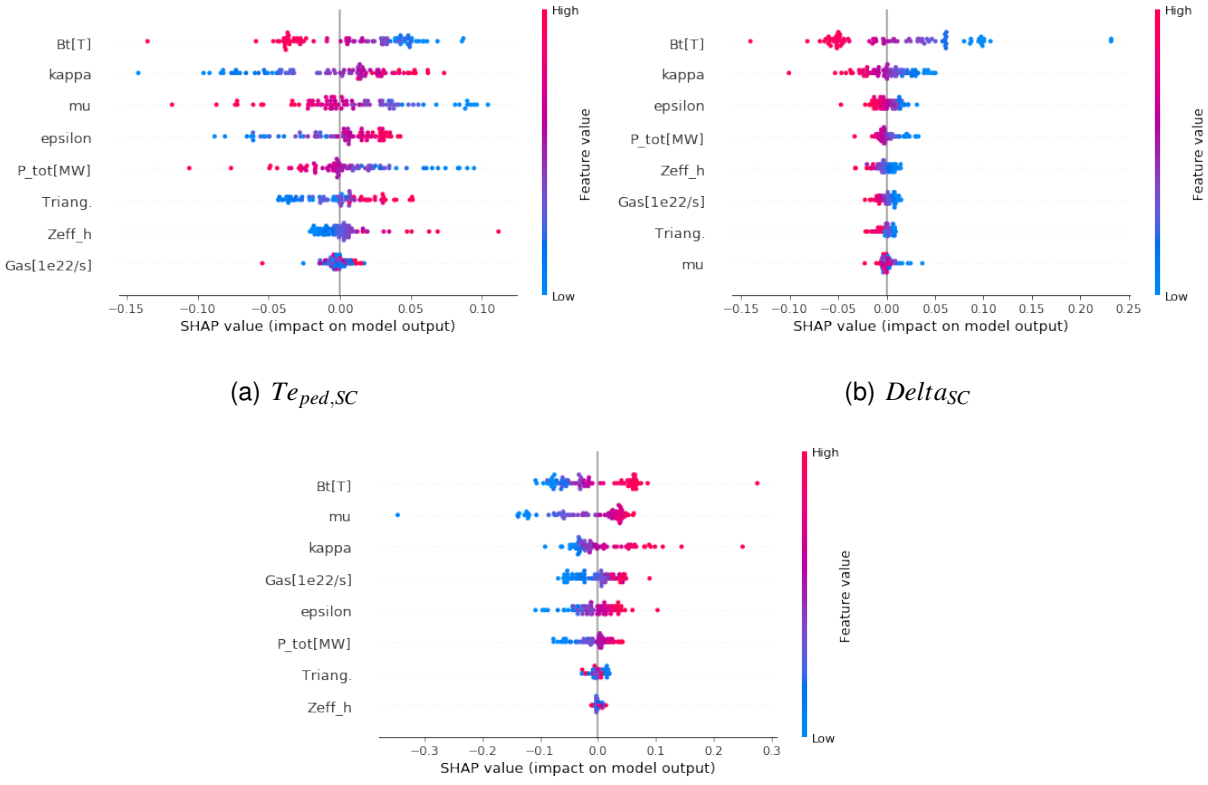
Figura 4.18: Incertidumbre relativa epistémica y aleatoria de las predicciones del modelo sustituto de EuroPED, para las variables β_p y Δ, en función de la distancia con respecto al centroide de los datos de entrenamiento.

Por consiguiente, en la figura 4.17, observamos distintos comportamientos. Tanto para las incertidumbres epistémicas como aleatorias de las predicciones de la variable β_p observamos que el error es bajo en la zona cercana al centroide y asciende a medida que nos alejamos del mismo, este es efectivamente el comportamiento esperado y que tanto hemos buscado. En las zonas de poca distancia al centroide se observa una pequeña sección con tendencia descendente, pero es debido al grupo de predicciones con valores cercanos al 0, tal y como explicamos en el anterior apartado. Sin embargo, para la variable Δ este comportamiento no sucede, la explicación a este suceso no es otra que la naturaleza de los propios datos. Es decir, tanto en los datos de entrenamiento como en los generados aleatoriamente, nos encontramos una variación en la variable Δ sensiblemente menor que en el caso de β_p, es por ello que la muestra no es igualmente representativa para ambas variables.

En el caso de la figura 4.18 se muestran los errores relativos (epistémicos y aleatorios) respecto de nuestras variables β_p y Δ, el crecimiento que se observa es menor, por tratarse del error relativo, pero aun así es apreciable.

A continuación, de la misma forma que en el modelo sustituto de EPED, incluimos los resultados que la herramienta SHAP arroja sobre las variables de entrada del modelo y su importancia. Hacemos nuevamente hincapié en la desconfianza que en ocasiones este valor tan informativo puede llegar a ocasionar debido a la variación entre modelos con diferentes parámetros de entrenamiento.

En este caso podemos observar como en las tres figuras 4.19(a), 4.19(b) y 4.19(c), el



(a) $Te_{ped,SC}$

(b) $Delta_{SC}$

(c) $ne_{ped,pred}$

Figura 4.19: Representación de la importancia de cada variable de entrada (de arriba a abajo, de más importante a menos) y su contribución a cada una de las variables de salida $Te_{ped,SC}$, $Delta_{SC}$ y $ne_{ped,pred}$, para un conjunto test de 84 entradas proveniente del dataset original, y para el modelo sustituto de EuroPED. Calculado mediante los valores SHAP. En rojo valores más altos de la variable de entrada y en azul los más bajos. Valor SHAP positivo significa que provoca que el valor de salida aumente, negativo que disminuya.

campo magnético toroidal $Bt[T]$ toma el papel predominante, lo que nos indica una mayor dependencia sobre el mismo en las tres variables de salida $Te_{ped,SC}$, $Delta_{SC}$ y $ne_{ped,pred}$. También vemos como las tendencias son más marcadas en el caso de $Te_{ped,SC}$ que en el resto. Del mismo modo la elongación del plasma *kappa* también toma un papel protagonista en las tres variables de salida.

Finalmente, podemos decir que hemos construido un modelo sustituto totalmente funcional para el complejo modelo EuroPED, habiendo comprobado que la distribución del error es la deseada o al menos se asemeja a esta. Del mismo modo que en EPED, se aprecia que el error aleatorio es mayor que el debido al propio modelo. Las ventajas de obtener un modelo sustituto son inmensas, ya que a través de este, la exploración de la física subyacente y la generación de datos, que solo podía llevarse a cabo experimentalmente, se agiliza a pasos agigantados. De hecho la construcción de modelos sustitutos o, en inglés, *surrogate modelling*, es cada vez más común en el campo de la fusión nuclear como se evidencia en [64].

4.3 Modelo Experimental y EuroPED

Tras haber realizado satisfactoriamente la implementación del método BNN-NCP, para construir modelos sustitutos de los modelos de pedestal de plasma EPED y EuroPED, procedemos a examinar los datos experimentales obtenidos del experimento JET-ILW [38]. Para ello desarrollaremos otro modelo, también basado en el método BNN-NCP, para capturar el comportamiento de estos datos experimentales. A fin de extraer conclusiones compararemos el modelo EuroPED con el modelo experimental.

En primer lugar, trataremos de observar el comportamiento de los datos experimentales en nuestra gráfica $\Delta - \beta_p$, para tener una primera aproximación de los mismos. Recordemos que para el cálculo de β_p son necesarias las variables Te_{ped}, ne_{ped}, $Triang.$, *rminor*, *kappa*, I_p y Z_{eff}. Por lo cual esta gráfica nos aporta información sobre gran cantidad de columnas. Pues bien, en la figura 4.20 se puede observar que no existe ninguna tendencia aparentemente en los datos experimentales, a diferencia del marcado caso de

EPED, y también de EuroPED donde la tendencia es más ligera pero existe. Claramente este conjunto de datos es independiente de la restricción $\Delta = c\sqrt{\beta_p}$ ya que el ajuste a la correspondiente curva roja es ridículamente inexacto.

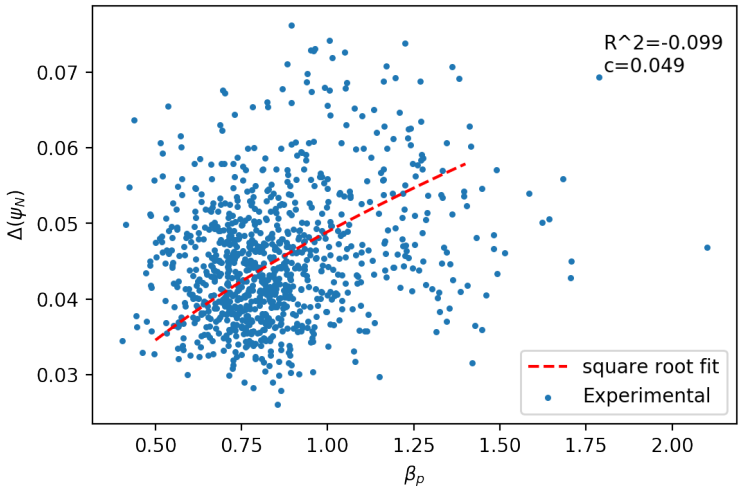


Figura 4.20: Gráfica $\Delta - \beta_p$ para los valores experimentales procedentes del conjuntos de datos JET, junto con su ajuste de raíz cuadrada.

4.3.1 Incógnita en EuroPED

Y bien, ¿qué clase de información podemos obtener a través de esta gráfica? Para responder a esta pregunta necesitamos volver al modelo EuroPED y a la figura 4.14 donde se representa la gráfica $\beta_p - \Delta$ del propio modelo. Como adelantábamos en la sección anterior, la dispersión de los datos en esa gráfica no se trataba de un detalle sin importancia. Este resultado planteó varias incógnitas en el transcurso de la investigación, y es que el modelo EuroPED contiene al modelo EPED y la restricción $\Delta = c\sqrt{\beta_p}$ se ha de cumplir, sin embargo las representaciones $\Delta - \beta_p$ no nos lo dejan tan claro.

Entonces, los datos hacían sospechar que existía una tercera variable involucrada que convertía la dispersión de los datos de la gráfica bidimensional, en un plano tridimensional, donde sí se cumple la restricción de raíz cuadrada para cada punto de la tercera variable. Encontrar este parámetro supuso realizar varias pruebas con diversas variables, pertenecientes al conjunto de datos o formadas por una combinación de las mismas. Si-

guiendo los resultados de la investigación [65], la primera variable propuesta fue pe_{ratio} descrita en la ecuación 4.1, que compara la presión del pedestal predicha por EuroPED y la experimental. Siendo q_e la carga del electrón, $ne_{ped,pred}$ y $Te_{ped,SC}$ la densidad electrónica y la temperatura del pedestal predichas por EuroPED, y las correspondientes a los datos experimentales $ne_{ped,exp}$ y $Te_{ped,exp}$.

$$pe_{ratio} = pe_{ped,crit} / pe_{ped,exp} \tag{4.1}$$

$$pe_{ped,crit} = q_e \cdot ne_{ped,pred} \cdot Te_{ped,SC}$$

$$pe_{ped,exp} = q_e \cdot ne_{ped,exp} \cdot Te_{ped,exp}$$

Sin embargo, como podemos apreciar en el mapa de colores de la figura 4.21, la tendencia no es tan clara para esta variable, lo cual se hace aún más evidente en una gráfica tridimensional, donde el plano buscado no se aprecia.

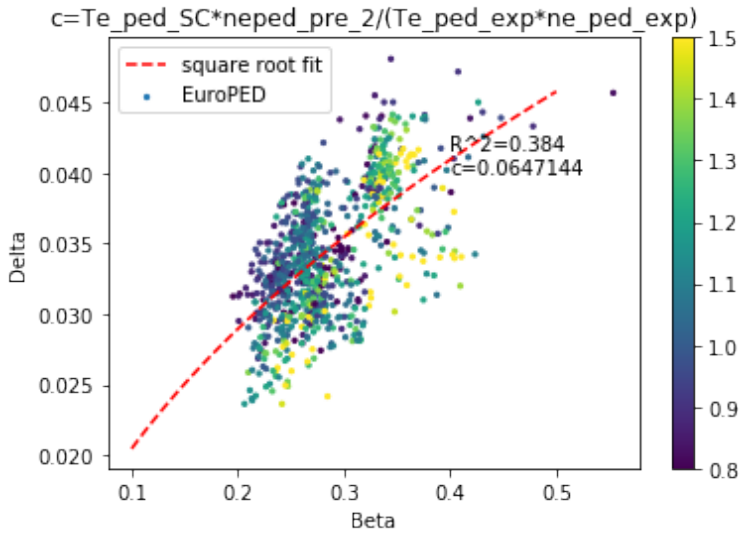


Figura 4.21: Datos de EuroPED en gráfica $\Delta - \beta_p$, con ajuste raíz cuadrada y mapa de color en función del valor de pe_{ratio} para cada punto.

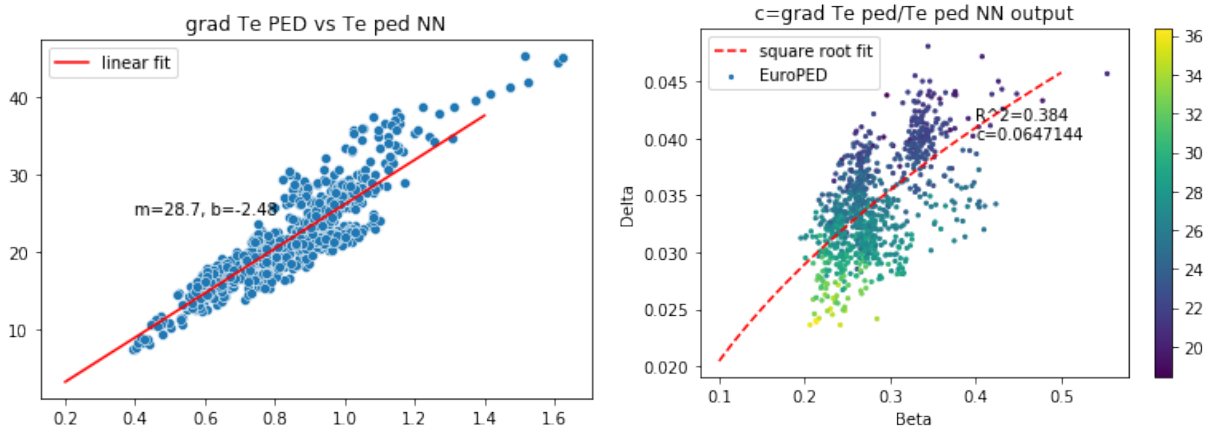
Dado este resultado, que aunque no fuera satisfactorio resulta interesante de comentar, habremos de buscar otra variable que funcione en nuestro cometido. Después de que mi supervisor Aaron contactase con otros investigadores que trabajaban con el tokamak ASDEX Upgrade, estos sugirieron que probáramos con la variable $grad\_Te_{ped}$ que no es

otra cosa que el gradiente del pedestal para el perfil de temperatura, por lo cual se define, tomando la temperatura en keV como en la ecuación 4.2.

$$grad\_Te_{ped} = \frac{Te_{ped} - 0.1}{\Delta} \tag{4.2}$$

Este valor comparado con Te_{ped} mostraba un comportamiento lineal en el tokamak ASDEX Upgrade, de esta forma lo tratamos de comprobar para nuestros datos experimentales procedentes del tokamak JET-ILW. Tras comprobar que efectivamente se mantenía el comportamiento lineal también en nuestros datos, verificamos que la tendencia se mantenía en la propia red neuronal (modelo sustituto de EuroPED) obteniendo como resultado la figura 4.22(a).

En la figura 4.22(b), utilizamos la variable $grad\_Te_{ped}/Te_{ped}$ construida mediante los



(a) $grad\_Te_{ped}$ vs Te_{ped}

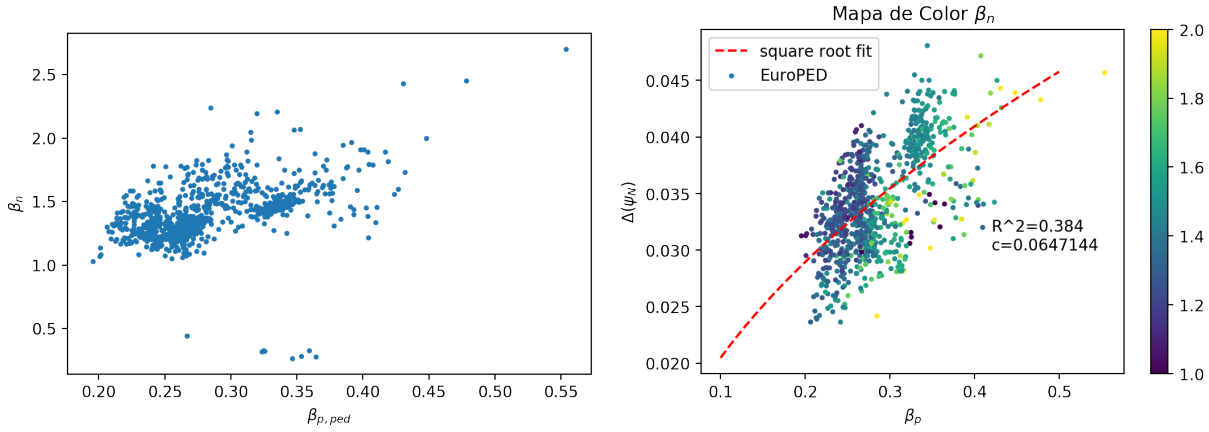
(b) $\Delta - \beta_p$ en función de $grad\_Te_{ped}/Te_{ped}$

Figura 4.22: A la izquierda, gráfica del gradiente de Te_{ped} en función de Te_{ped} (keV) para la salida de la red neuronal de EuroPED, con su ajuste lineal, donde m y b coinciden con el ajuste $y = mx + b$. A la derecha, los datos de EuroPED en gráfica $\Delta - \beta_p$, su ajuste raíz cuadrada y un mapa de color en función del valor de $grad\_Te_{ped}/Te_{ped}$ para la predicción de la red neuronal.

datos de la salida de la red neuronal para comprobar su dependencia en los datos de EuroPED. Como se aprecia, la tendencia es clara, sin embargo, cuando los datos son demasiado buenos es natural sospechar de los mismos. Así es como ocurre en este caso, esta tendencia tan marcada se debe efectivamente a que, para calcular β_p (componente horizontal en la figura 4.22(b)) estamos utilizando Te_{ped}, además Δ está presente en el cálculo de $grad\_Te_{ped}$. Por ello esta variable, a pesar de mostrar resultados deseables, no nos aporta

nueva información, ya que tan solo es una combinación de Δ y Te_{ped}, y por tanto no cumple su cometido.

De esta manera, para tratar de encontrar una variable adecuada con la que los datos de EuroPED formen un plano en la representación tridimensional $\beta_p - \Delta - z$ (siendo z la variable por determinar), y darle una explicación a la dispersión de los datos en la figura 4.14, correspondiente a la gráfica $\Delta - \beta_p$ de EuroPED, debemos de elegir una variable independiente de las variables Δ y β_p. Solo así nuestros resultados serán validos y aportarán un nuevo enfoque. Por lo que, tras probar con diferentes variables del conjunto de datos descrito en la sección 3.3, obtenemos los resultados más adecuados con la variable β_n que representa la β plasmática total definida en la ecuación 2.7. Recordemos que la variable β_p representa la contribución poloidal de la β total. Pues bien, en el conjunto de datos tenemos *beta_n_exp* como el valor de β total medido experimentalmente y *beta_n_pred* como la predicción original de EuroPED para la β total o global.



(a) β_n frente a β_p

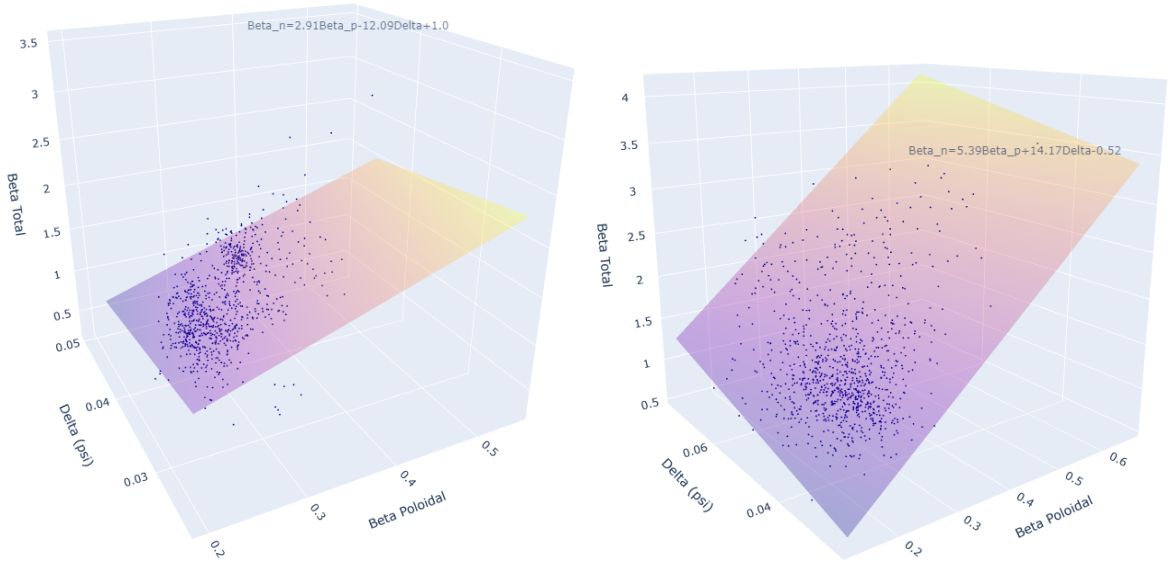
(b) $\Delta - \beta_p$ según β_n.

Figura 4.23: Representaciones de los datos de EuroPED. A la izquierda gráfica que compara β_n y $\beta_{p,ped}$. A la izquierda gráfica $\Delta - \beta_{p,ped}$ con ajuste a raíz cuadrada y mapa de colores en función del valor de β_n para cada dato.

Para demostrar la independencia de la variable β_n, tenemos la figura 4.23(a) donde se comparan las variables β_n (global) y β_p (poloidal). En ella vemos que la correlación entre ambas es vaga, por lo cual podemos decir que la variable β_n aporta información nueva.

Además de esto, la propia naturaleza de β_n, tal y como se define en la ecuación 2.7, provoca que sea una variable global para cada descarga del plasma ya que en su cálculo se promedia sobre todo el volumen, sin embargo este no es el caso de la variable β_p (definida en 2.6) la cual existe para cada superficie de flujo. Este hecho también contribuye a que β_n sea lo suficientemente independiente de β_p.

En la figura 4.23(b) podemos observar nuestro gráfico $\Delta - \beta_{p,ped}$ junto con un mapa de color según el valor de la variable β_n, es aquí donde podemos apreciar si el comportamiento que esperamos existe. Efectivamente se puede vislumbrar que dependiendo de las zonas dentro del espacio $\Delta - \beta_{p,ped}$ el valor de β_n es mayor o menor. Esto encaja con nuestra idea de que una tercera variable debería formar un plano en el espacio tridimensional para explicar la dispersión en los resultados de EuroPED.



(a) EuroPED (b) Experimental

Figura 4.24: Representaciones tridimensionales de los datos de EuroPED y experimentales, respectivamente, en el espacio $\beta_p - \Delta - \beta_n$, junto con el ajuste a un plano por mínimos cuadrados y la correspondiente ecuación del plano.

En la figura 4.24(a) podemos observar los datos para las predicciones de EuroPED representados en el espacio tridimensional $\beta_p - \Delta - \beta_n$. Efectivamente los datos se aproximan a un plano, con un cierto ruido, lo cual es esperable. Asimismo la ecuación del plano se muestra en la figura. Por ello queda confirmado que β_n es la tercera variable que necesitamos para explicar el hecho de que los datos de EuroPED no se ajusten completa-

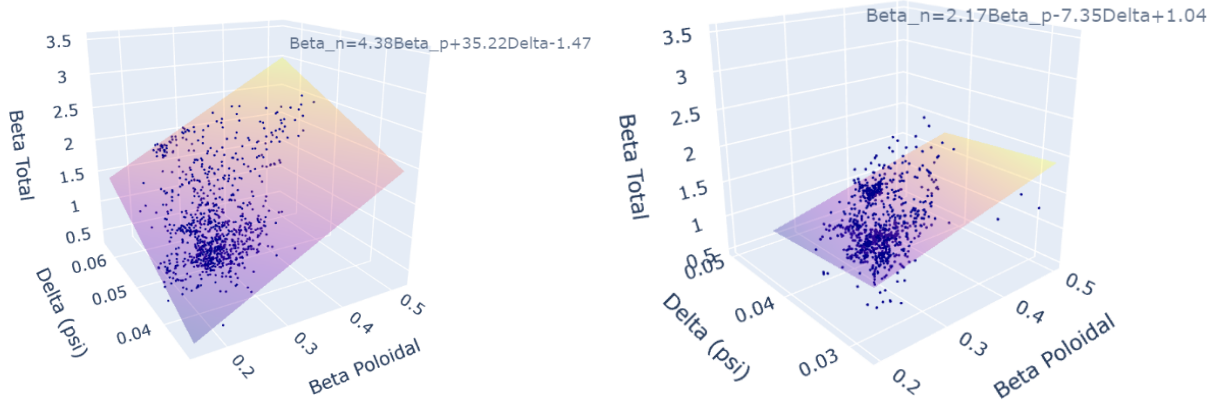
mente a la restricción $\Delta = c\sqrt{\beta_p}$. Es decir, debido a que se forma este plano, la restricción citada se cumple para cada valor de β_n, pero debido a que se representan todos estos valores de β_n en la representación bidimensional de la figura 4.14, es imposible distinguir con claridad esta tendencia en la misma. Esto plantea la interesante cuestión de por qué ocurre esto, y por qué en función de β_n. Aunque la respuesta está fuera del propósito de esta investigación, todavía podemos indagar un poco más.

4.3.2 Modelo Experimental y EuroPED con β_n

Como vaticinamos al inicio de esta sección, aunque en la representación de la figura 4.20, parezca que los datos experimentales no nos aportan ninguna información relevante, esto no resultará ser de esa forma. Ya que, tras observar el comportamiento de los datos de EuroPED con respecto a la variable β_n en la figura 4.24(a), decidimos hacer lo mismo con los datos experimentales. De este modo encontramos en la figura 4.24(b) una representación de los datos experimentales en el espacio $\beta_p - \Delta - \beta_n$, junto con su ajuste a un plano y la correspondiente ecuación del plano. En esta figura se evidencia que los datos experimentales, al igual que los datos de EuroPED ofrecen el mismo comportamiento respecto de la variable β_n, es decir forman un plano. Si bien es cierto que los planos que se forman para los datos de EuroPED y los experimentales son diferentes, en ambas ecuaciones del plano podemos observar que la relación de β_n con respecto a β_p es creciente, es decir su coeficiente es positivo, sin embargo esta relación no se mantiene para Δ donde el coeficiente tiene signo positivo (creciente) en el caso experimental y decreciente en EuroPED. De este modo se están poniendo en evidencia ciertos comportamientos del modelo EuroPED respecto de los datos experimentales, dejando claro que la influencia de la variable β_n es cuanto menos relevante.

Para corroborar este comportamiento, hemos construido un modelo sustituto empleando la misma técnica BNN-NCP para los datos experimentales, de manera que la red neuronal replicaría el comportamiento de la propia experimentación en el tokamak, lo que previsiblemente puede ofrecer información muy valiosa sobre la física subyacente. Para

ello se han utilizado las mismas variables de entrada que en el caso de EuroPED y las consecuentes variables de salida del caso experimental añadiendo la variable β_n, es decir, *Delta_exp*, *Te_ped_exp*, *ne_ped_exp* y *beta_n_exp*. De esta manera podremos comparar cómo el método BNN-NCP está replicando el comportamiento experimental mediante la gráfica tridimensional $\beta_p - \Delta - \beta_n$ comparándola con los datos reales mostrados en la figura 4.24(b).



(a) Experimental (b) EuroPED

Figura 4.25: Representaciones tridimensionales de las predicciones del modelo sustituto de EuroPED ampliado y experimental, respectivamente, en el espacio $\beta_p - \Delta - \beta_n$, junto con el ajuste a un plano por mínimos cuadrados y la correspondiente ecuación del plano.

De este modo, en la figura 4.25(a) podemos ver la predicción de la red en el espacio tridimensional $\beta_p - \Delta - \beta_n$ junto con su ajuste a un plano. Observamos cómo los coeficientes son similares a los de la figura 4.24(b) correspondientes a los datos experimentales, y además el comportamiento es apreciablemente similar en cuanto al ajuste al plano. Con este propósito queda definido el modelo experimental, de donde se pueden llevar a cabo gran cantidad de análisis, que exceden los propósitos de este proyecto, además de la generación de datos experimentales sintéticos para aumentar la cantidad de datos a utilizar en diversas investigaciones.

Con la conocida dependencia respecto de β_n, podemos expandir el modelo sustituto de EuroPED y tomar la variable predicción de EuroPED de β_n (*beta_n_pred*) como variable de salida, obteniendo un modelo más completo con respecto a las dependencias

del mismo y pudiendo comparar su resultado con los datos propios del modelo EuroPED dispuestos en la figura 4.24(a). Asimismo, obtenemos la predicción mediante el modelo sustituto ampliado y observamos el resultado dispuesto en el espacio tridimensional $\beta_p - \Delta - \beta_n$, en la figura 4.25(b). En ella podemos observar de nuevo la dependencia de β_n y el comportamiento planario con su correspondiente ajuste. Vemos también que los coeficientes son próximos a los del modelo original en la figura 4.24(a), lo que corrobora que la ampliación de modelo sustituto de EuroPED con la variable de salida β_n posee el comportamiento deseado.

Para terminar, realizaremos una comprobación de la ya conocida dependencia de EuroPED con respecto a la variable β_n. Sabemos que nuestro modelo sustituto, además de replicar el comportamiento, también es capaz de captar las dependencias y trabajar con las mismas, por ello, aunque introdujéramos datos de entrada aleatorios, nuestro modelo debería mantener ciertos comportamientos intrínsecos. De esta manera, hemos tomado datos de entrada aleatorios dentro del mismo rango con el que se generaron las gráficas en las figuras 4.16(a) y 4.16(b), elaboradas también en el marco del modelo sustituto de EuroPED. Sin embargo, en este caso también vamos a obtener información sobre β_n en la salida de la red neuronal. Por todo esto, la predicción del modelo sustituto ampliado de EuroPED sobre los datos aleatorios se encuentra en la figura 4.26 representada en nuestro conocido espacio tridimensional. En el gráfico, se puede ver que hay una tendencia muy aproximada que sigue el ajuste planario. Sin embargo, la dispersión de los datos es muy grande, por lo que no podemos afirmarlo con certeza. En cualquier caso, el resultado no es concluyente por el momento. Por lo tanto, esto requiere un trabajo futuro para poder decir algo más sólido y concreto sobre la dependencia de β_n, lo cual queda fuera del objetivo del proyecto. Aun así es claro que la dependencia existe.

Como punto final, si recordamos cuando definimos y detallamos el modelo EuroPED en la sección 2.2.2, se usaba para estimar la β global el modelo de transporte *Bohm-gyroBohm* [42]. En una nota al pie aclarábamos que aunque en la actualidad existen modelos mucho más precisos que tienen en cuenta el transporte turbulento dentro del núcleo, este proporciona una buena estimación de orden cero a efectos de predecir la beta global. Dicho esto, podrían existir discrepancias entre la beta predicha por este método y la beta experimental. Esto último puede observarse claramente en las figuras 4.24(b) y 4.24(a) y

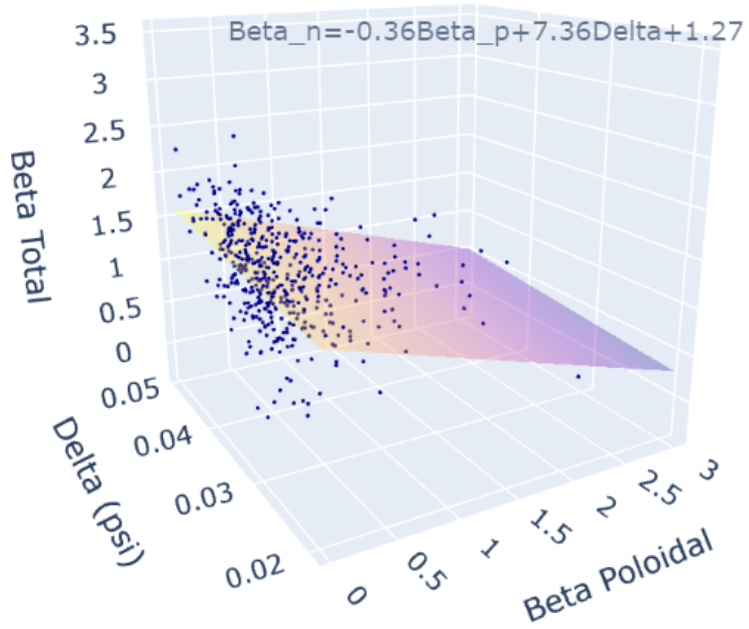


Figura 4.26: Representación tridimensional de la predicción del modelo sustituto de EuroPED ampliado para datos de entrada aleatorios en el espacio $\beta_p - \Delta - \beta_n$

en las discrepancias en su plano de ajuste. Como sabemos, estos gráficos tridimensionales muestran que hay una dependencia adicional de la β global (β_n) en los datos, la cual no es capturada por EPED. Si bien es cierto que EuroPED muestra esta variabilidad entre la predicción y el valor real de β_n, no sabemos hasta qué punto esta aproximación (el propio modelo *Bohm-gyroBohm*) afecta a la precisión de nuestros resultados. Una futura expansión del proyecto podría ser utilizar un modelo de transporte más preciso en EuroPED, para verificar de esta forma si la dependencia de β_n está adecuadamente cubierta ya por EuroPED o es un resultado de la física que falta en el modelo EPED.

Capítulo 5

Conclusiones

Para terminar, en este capítulo se realizará un sumario sobre las conclusiones citadas a lo largo del trabajo. De este modo se ofrecerá una visión general de los avances científicos que han tenido lugar en la investigación.

Primeramente, destacar que los objetivos principales han sido completados. Es decir, hemos logrado elaborar el método de red neuronal bayesiana con constraste de ruido y construir con él los modelos sustitutos de los modelos de pedestal de plasma EPED y EuroPED.

En primera instancia, ampliamos el método BNN-NCP para dos dimensiones de entrada y tomamos valores de nuestro conjunto de datos para realizar algunas predicciones y comprobar el funcionamiento del método. Dado que los resultados fueron satisfactorios pasamos a replicar modelos de pedestal de plasma completos.

5.1 EPED

En el caso de EPED, tal y como hemos detallado en las secciones 2.2.1 y 2.2.2, conocíamos que era el más sencillo de los dos modelos. Aún así, la implementación del método BNN-NCP en la elaboración del modelo sustituto no era una tarea sencilla. No obstante, los resultados y la eficacia del modelo sustituto han resultado sorprendentemen-

te positivos. Recordemos que durante todo el análisis de resultados hemos hecho uso de la gráfica que compara la β_p poloidal, calculada haciendo uso de las predicciones del modelo, y la anchura del pedestal Δ, que constituye en sí misma una variable de predicción del modelo. Observamos la gráfica $\beta_p - \Delta$ para EPED en la figura 4.6.

En primer lugar, el modelo construido aproxima la constante c de la relación $\Delta = c\sqrt{\beta_p}$ (que esperamos que se cumpla en EPED) con una precisión muy alta: si el valor de c para las predicciones del modelo original EPED es de $c_{EPED} = 0.06212$, el valor para las predicciones de nuestro modelo ha resultado ser de $c_{Model,EPED} = 0.06222$. Además, el ajuste a la función raíz cuadrada resulta altamente satisfactorio. Lo podemos encontrar en las figuras 4.7(a) y 4.7(b). La incertidumbre epistémica (incertidumbre debida al modelo) resulta considerablemente baja en las predicciones, sobre todo si la comparamos con la incertidumbre aleatoria (incertidumbre provocada por el ruido subyacente a los datos de entrada).

Asimismo, cuando realizamos nuestras predicciones del modelo sustituto de EPED sobre datos aleatorios, encontramos que las predicciones se agrupan en la zona donde la red ha entrenado y está acostumbrada a trabajar, ciñéndose a la restricción de la raíz cuadrada, lo cual es una señal de que está realizando su tarea de manera adecuada. Lo podemos observar en las figuras 4.8(a) y 4.8(b). Del mismo modo, uno de los resultados más esperados del modelo era ver cómo los errores de las predicciones aumentan (al menos en el caso del epistémico) en las zonas donde el modelo no ha recibido datos de entrenamiento, por lo que está extrapolando. Esto se cumple muy claramente para las variables β_p y Δ en la figura 4.9, donde vemos que los errores crecen con la distancia al centroide de los datos.

5.2 EuroPED

En cuanto al modelo EuroPED, el cual se describe con detalle en la sección 2.2.2, se trata de un modelo de pedestal de plasma de una complejidad sensiblemente superior al modelo EPED, de hecho EuroPED contiene directamente a EPED. Por ser un modelo más complejo también es esperablemente más preciso y contiene más información sobre

el plasma que modela. Por ello a nivel de estudio, es un modelo mucho más interesante. Sin embargo, vemos que nuestro método sigue capturando el comportamiento de una manera muy eficaz si comparamos las figuras 4.14 y 4.15(a). Lo hace con un error epistémico y aleatorio (figura 4.15(b)) un tanto mayor que en el caso de EPED pero bajo valores razonables. En el caso de los datos de entrada aleatorios, como observamos en las figuras 4.16(a) y 4.16(b), las predicciones se distribuyen en torno a la zona de entrenamiento, pero con una dispersión significativamente mayor, pues la dispersión de las predicciones de EuroPED en el espacio $\beta_p - \Delta$ es a su vez mayor. En cuanto al comportamiento del modelo sustituto respecto a la incertidumbre, en la figura 4.17 vemos claramente como respecto a la variable β_p el comportamiento es el esperado, es decir, ambos errores aumentan según nos alejamos de la zona de entrenamiento. Sin embargo, para la variable Δ la tendencia existe pero no es tan clara, debido principalmente a la falta de variación de los datos en la variable Δ, lo cual está íntimamente relacionado con la naturaleza del modelo.

Tanto en el caso del modelo sustituto de EPED como de EuroPED, se ha aplicado la técnica SHAP que nos ofrece la importancia y el efecto de cada variable de entrada en la predicción final. Sin embargo, lo que podría ser una potente herramienta finalizó demostrando menos robustez de la esperada. Por ello, no concluiremos resultados absolutos sobre la influencia de las variables, a pesar de que en gran parte de los casos el efecto de las variables de entrada que se muestra en los diagramas SHAP (figuras 4.11(b), 4.11(a), 4.19(b), 4.19(c), 4.19(a)) se encuentra en consonancia con la física subyacente conocida sobre el sistema.

Finalmente, podemos decir que hemos construido dos modelos sustitutos totalmente funcionales para los modelos EPED y EuroPED, lo cual completa el objetivo fundamental del proyecto. Además, estos modelos poseen el comportamiento deseado respecto a los errores epistémico y aleatorio. Es decir, aumenta generalmente en las zonas alejadas de los datos de entrenamiento, donde el modelo extrapola los resultados, dado que no tiene información especifica sobre esa región.

5.3 Dependencia de β_n y Modelo experimental

Tras advertir la dispersión de las predicciones de EuroPED respecto de la restricción $\Delta = c\sqrt{\beta_p}$, la cual supuestamente debería de cumplirse debido a que EPED está contenido en EuroPED, se ha llevado a cabo un análisis deductivo. En este se ha vislumbrado que existe una tercera variable que aportaría más luz al asunto. Tras probar con diferentes variables, finalmente se ha concluido que se trata de la β plasmática total, denominada β_n. En la figura 4.23(a) hemos podido comprobar que la independencia con respecto a β_p es suficiente como para decir que esta variable proporciona información nueva, además en la figura 4.23(b) corroboramos que el comportamiento del espacio $\Delta - \beta_{p,ped}$ es el esperado con respecto a $\beta_{n,ped}$. Esta variable provoca la dispersión bidimensional de las predicciones en torno al ajuste de raíz cuadrada que apreciamos en la figura 4.14. Tal como se aprecia en la figura 4.24(a), existe una clara tendencia coplanaria en el espacio tridimensional $\beta_p - \Delta - \beta_n$, es decir, la dispersión del espacio bidimensional está provocada por la dependencia de la variable β_n.

Del mismo modo, comprobamos que efectivamente la tendencia sigue existiendo en los datos experimentales (figura 4.24(b)), lo cual es una señal de que este comportamiento es propio del experimento, no solo del modelo EuroPED. Aprovechamos este hecho para expandir el modelo sustituto de EuroPED con la predicción de la variable β_n para de este modo poder captar y visualizar un mayor número de dependencias. Por ello en la figura 4.25(b), se encuentran las predicciones del modelo sustituto expandido de EuroPED, las cuales se asemejan considerablemente al modelo original en la figura 4.24(a). Para obtener más información sobre la dependencia de β_n en EuroPED, se llevan a cabo predicciones de datos aleatorios mediante el modelo sustituto expandido, y se comprueba en la figura 4.26 que a pesar de que la dependencia existe, el plano que se forma es tan disperso que no nos aventuramos a obtener conclusiones del mismo. Es decir, sería necesario un trabajo futuro para obtener más información del sistema, lo cual se aleja de los objetivos de este proyecto.

Asimismo, aprovechando la dependencia de β_n en los datos experimentales y la ma-

duración de nuestro método BNN-NCP utilizado a lo largo de todo el proyecto, decidimos construir un modelo sustituto de los propios datos experimentales, que en este caso replicaría el funcionamiento del propio tokamak. Esta es una herramienta muy potente con la que nos hemos limitado a construir y comprobar su rendimiento, tal y como podemos observar si comparamos las figuras 4.24(b) y 4.25(a). Esta herramienta abre futuras vías de investigación en la exploración del pedestal del plasma en procesos de fusión nuclear, pues se ha conseguido elaborar un modelo sustituto del propio tokamak, un dispositivo que requiere de grandes cantidades de energía para su funcionamiento. Esto es una gran ventaja tanto a la hora de generar datos para alimentar otras investigaciones, como en el caso de explorar el propio comportamiento del plasma y la física subyacente.

Por último, hemos terminado proponiendo, como futuros pasos, la sustitución del modelo de transporte *Bohm-gyroBohm* para estimar la β global en el modelo EuroPED, por uno más preciso, para de esta forma poder verificar si la dependencia de β_n ya está adecuadamente cubierta por EuroPED o se trata de un resultado de la física que falta en el modelo EPED.

Además de las ventajas prácticas que puede presentar la metodología planteada, las cuales han sido expuestas hasta aquí, cabe destacar otra aportación de este trabajo. En el mismo se combinan conceptos clásicos de matemáticas, que son aplicados a nuevos métodos computacionales, que a su vez permiten resolver un problema físico. Así pues, es posible ver como de un estudio matemático riguroso se puede llegar a interesantes aplicaciones en la física. La combinación de ambos puntos de vista ha sido un reto, así como el desarrollo del código necesario para ello, pero es un ejemplo adecuado de las sinergias entre ambos grados académicos.

Además de esto, a nivel personal me ha ayudado a decantar mi futuro hacia la rama que este trabajo representa, la intersección entre el *machine learning* y la fusión nuclear. De este modo el desempeño de estos 5 años en ambos grados cobra sentido, consiguiendo articular ambas disciplinas para un mismo fin.

Finalmente, comentar que a lo largo del verano del 2022 se elaborará el manuscrito pertinente a este proyecto para ser enviado a alguna de las revistas científicas centradas en física de plasmas y fusión nuclear.

Bibliografía

[1] BP, «BP Statistical Review of World Energy», 68th ed., acceso 22/03/2022, **2019**. 1.1.1

[2] IPCC. «Climate Change 2001: The Scientific Basis. Contribution of Working Group I to the Third Assessment Report of the Intergovernmental Panel on Climate Change», accesso 22/03/2022, **2001**. 1.1.1

[3] J. Wesson. «Tokamaks», 3rd ed., *Oxford University Press*, **2004**. 2.1

[4] CIEMAT. «Flux Surface», *FusionWiki*. Última modificación **19 Julio de 2012**. http://fusionwiki.ciemat.es/wiki/Flux_surface. 2.1.1

[5] S.S. Abdullaev, K.H. Finken, M. Jakubowski y M. Lehnen. «Mappings of stochastic field lines in poloidal divertor tokamaks». *Nucl. Fusion*, **2006**, 46, S113. 2.3

[6] M. S. Chu y M. Okabayashi. «Stabilization of the external kink and the resistive wall mode», *Plasma Phys. Control. Fusion* **2010**, 52, 123001. 2.1.2

[7] E. Morse. *Magnetohydrodynamic Stability*. En *Nuclear fusion*. Springer, **2019**. 2.1.2

[8] F. Wagner et al. «Regime of Improved Confinement and High Beta in Neutral-Beam-Heated Divertor Discharges of the ASDEX Tokamak», *Phys. Rev. Lett.* **1982**, 49, 1408. 2.4, 2.1.2

[9] CIEMAT. «Pedestal», *FusionWiki*, ultima modificación **3 abril de 2018**, http://fusionwiki.ciemat.es/wiki/Pedestal 2.5

[10] P. Gohil, M. Ali Mahdavi, L. Lao, K. H. Burrell, M. S. Chu, J. C. DeBoo, C. L. Hsieh, N. Ohyabu, R. T. Snider, R. D. Stambaugh, y R. E. Stockdale. «Study of

Giant Edge-Localized Modes in DIII-D and Comparison with Ballooning Theory», *Phys. Rev. Lett.* **1988**, 61, 1603. 2.1.2

[11] K. H. Burrell, M. E. Austin, D. P. Brennan, J. C. DeBoo, E. J. Doyle, P. Gohil, C. M. Greenfield, R. J. Groebner, L. L. Lao, T. C. Luce, M. A. Makowski, G. R. McKee, R. A. Moyer, T. H. Osborne, M. Porkolab, T. L. Rhodes, J. C. Rost, M. J. Schaffer, B. W. Stallard, E. J. Strait, M. R. Wade, G. Wang, J. G. Watkins, W. P. West y L. Zeng. «Quiescent H-mode plasmas in the DIII-D tokamak», *Plasma Phys. Control. Fusion* **2002**, 44, A253. 2.1.2

[12] ASDEX Team. «The H-mode of ASDEX», *Nucl. Fusion* 29, 1959 (**1989**). 2.1.2

[13] H. Zohm. «Edge localized modes (ELMs)». *Plasma Phys. Controll. Fusion* 38(2), 105 **1996**. 2.1.2

[14] P. B. Snyder, H. R. Wilson, J. R. Ferron, L. L. Lao, A. W. Leonard, T. H. Osborne, A. D. Turnbull, D. Mossessian, M. Murakami y X. Q. Xu. «Edge localized modes and the pedestal: A model based on coupled peelingballooning modes», *Phys. Plasmas* **2002**, 9, 2037. 2.1.2, 2.1.2, 2.2.2

[15] A. B. Mikhailovskii, G. T. A. Huysmans, W. O. K. Kerner, y S. E. Sharapov. «Optimization of computational MHD normal-mode analysis for tokamaks», *Plasma Phys. Rep.* **1997**, 23, 844. 2.1.2, 2.2.2

[16] R. L. Miller, M. S. Chu, J. M. Greene, Y. R. Lin-Liu y R. E. Waltz. «Noncircular, finite aspect ratio, local equilibrium model», *Phys. Plasmas* **1998**, 5, 973. 2.1.2

[17] Nyström, H. (**2020**). «Modeling of the pedestal performance in the DTT tokamak». Tesis de Máster, Universidad de Uppsala. 2.6, 2.11, 2.12

[18] O. Sauter, C. Angioni y Y. R. Lin-Liu. «Neoclassical conductivity and bootstrap current formulas for general axisymmetric equilibria and arbitrary collisionality regime», *Phys. Plasmas* **1999**, 6, 2834. 2.1.2

[19] C. Perez von Thun, L. Frassinetti, L. Horvath, S. Saarelma, L. Meneses, E. de la Luna, M. Beurskens, J. Boom, J. Flanagan, J.C. Hillesheim, C.F. Maggi, S.J.P. Pamela,

E.R. Solano y *JET Contributors*. «Long-lived coupled peeling ballooning modes preceding ELMs on JET», *Nucl. Fusion* **2019**, 59, 056004. 2.1.2, 2.1.2

[20] A. Loarte, M. Becoulet, G. Saibene, R. Sartori, D. J. Campbell, T. Eich, A. Herrmann, M. Laux, W. Suttrop, B. Alper, P. J. Lomas, G. Matthews, S. Jachmich, J. Ongena, P. Innocente y *EFDA- JET Workprogramme Collaborators*. «Characteristics and scaling of energy and particle losses during Type I ELMs in JET H-modes», *Plasma Phys. Control. Fusion* **2002**, 44, 1815. 2.1.2

[21] R.J. Groebner, D.R. Baker, K.H. Burrell, T.N. Carlstrom, J.R. Ferron, P. Gohil, L.L. Lao, T.H. Osborne, D.M. Thomas, W.P. West, J.A. Boedo, R.A. Moyer, G.R. McKee, R.D. Deranian, E.J. Doyle, C.L. Rettig, T.L. Rhodes y J.C. Rost. «Progress in quantifying the edge physics of the H mode regime in DIII-D», *Nuclear Fusion* **2001**, 41, 1789. 2.1.2, 2.7

[22] R. J. Groebner, T. H. Osborne.«Scaling studies of the high mode pedestal», *Phys. Plasmas*. **1998**,5, 1800. 2.1.2

[23] William Denis Dhaeseleer, William Nicholas Guy Hitchon, James D. Callen y J. Leon Shohet. «Flux Coordinates and Magnetic Field Structure», *Springer*, **1991**. 2.1.2

[24] M. N. A. Beurskens, J. Schweinzer, C. Angioni, A. Burckhart, C. D. Challis, I. Chapman, R. Fischer, J. Flanagan, L. Frassinetti, C. Giroud, J. Hobirk, E. Joffrin, A. Kallenbach, M. Kempenaars, M. Leyland, P. Lomas, G. Maddison, M. Maslov, R. McDermott, R. Neu, I. Nunes, T. Osborne, F. Ryter, S. Saarelma, P. A. Schneider, P. Snyder, G. Tardini, E. Viezzer, E. Wolfrum, *the ASDEX Upgrade Team* y *JET-EFDA Contributors*. «The effect of a metal wall on confinement in JET and ASDEX Upgrade», *Plasma Phys. Control. Fusion* **2013**, 55, 124043. 2.8

[25] P. B. Snyder, R. J. Groebner, A. W. Leonard, T. H. Osborne y H. R. Wilson. «Development and validation of a predictive model for the pedestal height», *Phys. Plasmas* **2009**, 16, 056118. 2.2.1, 2.9, 2.2.1, 2.10, 2.2.2, 4.2.1

[26] J. W. Connor, R. J. Hastie, H. R. Wilson y R. L. Miller. «Magnetohydrodynamic stability of tokamak edge plasmas», *Phys. Plasmas* 5, 2687 **1998**. 2.2.1

[27] H. R. Wilson, P. B. Snyder y R. L. Miller. «Numerical studies of edge localized instabilities in tokamaks», *Phys. Plasmas* 9, 1277 **2002**. 2.2.1, 2.2.2

[28] P. B. Snyder, H. R. Wilson, J. R. Ferron, L. L. Lao, A. W. Leonard, T. H. Osborne, A. D. Turnbull, D. Mossessian, M. Murakami, y X. Q. Xu. «Edge localized modes and the pedestal: A model based on coupled peelingballooning modes», *Phys. Plasmas* 9, 2037 **2002**. 2.2.1

[29] G. T. A. Huysmans. «ELMs: MHD instabilities at the transport barrier», *Plasma Phys. Controlled Fusion* 47, B165 **2005**. 2.2.1

[30] H. R. Wilson, S. C. Cowley, A. Kirk, y P. B. Snyder. «Magneto-hydrodynamic stability of the H-mode transport barrier as a model for edge localized modes: an overview», *Plasma Phys. Controlled Fusion* 48, A71 **2006**. 2.2.1

[31] T. H. Osborne, K. H. Burrell, R. J. Groebner, L. L. Lao, A. W. Leonard, R. Maingi, R. L. Miller, G. D. Porter y A. D. Turnbull. «H-mode pedestal characteristics in ITER shape discharges on DIII-D», *J. Nucl. Mater.* 266269, 131 **1999**. 2.2.1

[32] R. J. Groebner y T. H. Osborne. «Scaling studies of the high mode pedestal», *Phys. Plasmas* 5, 1800. **1998**. 2.2.1

[33] P. B. Snyder, N. Aiba, M. Beurskens, R. J. Groebner, L. D. Horton, A. E. Hubbard, J. W. Hughes, G. T. A. Huysmans, Y. Kamada, A. Kirk, C. Konz, A. W. Leonard, J. Lönnroth, C. F. Magii, R. Maingi, T. H. Osborne, N. Oyama, A. Pankin, S. Saarelma, G. Saibene, J. L. Terry, H. Urano y H. R. Wilson. «Pedestal stability comparison and ITER pedestal prediction», *Proceedings of the 22nd International Conference*, Geneva, Switzerland, **2008**, Paper No. IT/P6-14, *Nucl. Fusion* submitted. 2.2.1

[34] S. Saarelma, A. Järvinen, M. Beurskens, C. Challis, L. Frassinetti, C. Giroud, M. Groth, M. Leyland, C. Maggi, J. Simpson y *JET Contributors*. «The effects of impurities and core pressure on pedestal stability in Joint European Torus (JET)». **2015** *Phys. Plasmas* 22 056115. 2.2.2

[35] M.J. Leyland, M.N.A. Beurskens, L. Frassinetti, C. Giroud, S. Saarelma, P.B. Snyder, J. Flanagan, S. Jachmich, M. Kempenaars, P. Lomas, G. Maddison, R. Neu, I. Nunes, K.J. Gibson2 y *JET-EFDA Contributors*. «The H-mode pedestal structure and its role on confinement in JET with a carbon and metal wall», **2015** *Nucl. Fusion* 55 013019. 2.2.2

[36] P.B. Snyder, R.J. Groebner, J.W. Hughes, T.H. Osborne, M. Beurskens, A.W. Leonard, H.R. Wilson y X.Q. Xu. «A first-principles predictive model of the pedestal height and width: development, testing and ITER optimization with the EPED model», **2011** *Nucl. Fusion* 51 103016. 2.2.2

[37] H. Urano. «Characterization of electron density based on operational parameters in JET H-mode plasmas with C and ILW» **2016** *Proc. 43rd EPS Conf. on Plasma Physics* (Leuven, Belgium, 48 July, 2016) O4.121. 2.2.2

[38] S. Saarelma, C. D. Challis, L. Garzotti, L. Frassinetti, C. F. Maggi, M. Romanelli, C. Stokes y *JET Contributors*. «Integrated modelling of H-mode pedestal and confinement in JET-ILW», **2018** *Plasma Phys. Control. Fusion* 60 014042. 2.2.2, 2.2.2, 2.13, 3.3, 4.3

[39] R. J. Groebner, M. A. Mahdavi, A. W. Leonard, T. H. Osborne, G. D. Porter, R. J. Colchin y L. W. Owen. «The role of neutrals in high-mode (H-mode) pedestal formation», **2002** *Phys. Plasmas* 9 2134. 2.2.2

[40] L. Frassinetti, S. Saarelma, P. Lomas, I. Nunes, F. Rimini, M. N. A. Beurskens, P. Bilkova, J. E. Boom, E. de la Luna, E. Delabie, P. Drewelow, J. Flanagan, L. Garzotti, C. Giroud, N. Hawks, E. Joffrin, M. Kempenaars, Hyun-Tae Kim, U. Kruezi, A. Loarte, B. Lomanowski, I. Lupelli, L. Meneses, C. F. Maggi, S. Menmuir, M. Peterka, E. Rachlew, M. Romanelli, E. Stefanikova y *JET Contributors*. «Dimensionless scalings of confinement, heat transport and pedestal stability in JET-ILW and comparison with JET-C», **2017** *Plasma Phys. Control. Fusion* 59 014014. 2.2.2

[41] D. Kim, A. Merle, O. Sauter y T. P. Goodman. «Simple predictive electron transport models applied to sawtoothing plasmas», **2016** *Plasma Phys. Control. Fusion* 58 055002. 2.2.2

[42] M. Erba, A. Cherubini, V. V. Parail, E. Springmann y A. Taroni. «Development of a non-local model for tokamak heat transport in L-mode, H-mode and transient regimes», **1997** *Plasma Phys. Control. Fusion* 39 261. 2.2.2, 4.3.2

[43] C. Angioni, H. Weisen, O.J.W.F. Kardaun, M. Maslov, A. Zabolotsky, C. Fuchs, L. Garzotti, C. Giroud, B. Kurzan, P. Mantica, A.G. Peeters, J. Stober1, *the ASDEX Upgrade Team* y *contributors to the EFDA-JET Workprogramme*. «Scaling of density peaking in H-mode plasmas based on a combined database of AUG and JET observations», **2007** *Nucl. Fusion* 47 1326. 2.2.2

[44] O. Sauter, C. Angioni y Y. R. Lin-Liu. «Neoclassical conductivity and bootstrap current formulas for general axisymmetric equilibria and arbitrary collisionality regime», **1999** *Phys. Plasmas* 6 2834. 2.2.2

[45] R. Hager y C. S. Chang. «Gyrokinetic neoclassical study of the bootstrap current in the tokamak edge pedestal with fully non-linear Coulomb collisions», **2016** *Phys. Plasmas* 23 042503. 2.2.2

[46] Wang, J. Rao y T. Lim. «Deep learning applications in medical image analysis», *IEEE Access*, vol. 6, pp. 93759389, **2018**. 2.3

[47] Q. Rao y J. Frtunikj. «Deep learning for self-driving cars: Chances and challenges», *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, ser. SEFAIS 18, **2018**, pp. 3538. 2.3

[48] Iberdrola. *Innovación*, último acceso 25/04/2022. https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico#:~:text=$%$20Machine$%$20Learning$%$20es$%$20una,elaborar$%$20predicciones$%$20(an$%$C3$%$A1lisis$%$20predictivo) 2.3

[49] N. Suga. «Biosonar and neural computation in bats», *Scientific American*. **1990**,a,b. 2.3.1

[50] S. Haykin, **1999**, «Introduction» en «Neural Network, A Comprehensive Foundation», 2nd ed. *Pearson Education*. 2.3.1, 2.14

[51] X. Sarasola. (**2021**). «Redes Neuronales: Taxonomía y bases matemáticas». Tesis de Fin de Grado, Universidad de Oviedo. 2.3.1

[52] I. Alvarado, «Redes Neuronales», *GitHub*. https://ml4a.github.io/ml4a/es/neural_networks/. 2.3.1

[53] E. Kreyszig, «Introductory Functional Analysis with Applications», *John Wiley*, **1991**. 2.3.1

[54] M. Hassoun, *M. Fundamentals of Artificial Neural Networks*. MIT Press. Cambridge, **1995**. 2.4

[55] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow y R. Fergus. «Intriguing properties of neural networks», *arXiv preprint* arXiv:1312.6199, **2013**. 2.4

[56] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine y M. Bennamoun. «Hands-on Bayesian Neural Networks – a Tutorial for Deep Learning Users», *arXiv preprint* arXiv:2007.06823v3, **2022** 2.4.1

[57] A. Etz, Q. F. Gronau, F. Dablander, P. A. Edelsbrunner y B. Baribault. «How to become a Bayesian in eight easy steps: An annotated reading list», *Psychonomic Bulletin Review*, vol. 25, pp. 219234, **2018**. 2.4.1

[58] C. Blundell. «Weight Uncertainty in Neural Networks», *arXiv preprint*, arXiv:1505.05424v2, **2015**. 2.4.2

[59] G. Mateo-García. «Bayesian neural networks». https://www.uv.es/gonmagar/blog/2018/03/15/BayesianNeuralNetworks. Último acceso: 03/05/2022. 2.4.3

[60] D. Hafner, D. Tran, T. Lillicrap, A. Irpan y J. Davidson. «Noise contrastive priors for functional uncertainty», *Uncertainty in Artificial Intelligence*, pages 905-914. PMLR, **2020**. 2.5, 2.5, 2.5, 4.1.1, 5.3

[61] G.J.E. van Otterdijk (**2021**). «Bayesian Neural Networks as uncertainty estimators». Tesis de Fin de Grado, Eindhoven University of Technology. 2.5.1, 2.17, 2.18

[62] R. Gupta. «Kl divergence between 2 gaussian distributions». https://mr-easy.github.io/2020-04-16-kl-divergence-between-2-gaussian-distributions/. Último acceso 07/05/2022. 3.1

[63] C. Kuo. «Explain Your Model with the SHAP Values». *Medium*. https://towardsdatascience.com/explain-your-model-with-the-shap-values-bc36aac4de3d. Último acceso 06/04/2022. 4.2.1

[64] P. Rodriguez-Fernandez, N.T. Howard y J. Candy, «Nonlinear gyrokinetic predictions of SPARC burning plasma profiles enabled by surrogate modeling», *Nucl. Fusion* 62, 076036 (**2022**). 4.2.2

[65] L. Frassinetti, S. Saarelma, G. Verdoolaege, M. Groth, J.C. Hillesheim, P. Bilkova, P. Bohm, M. Dunne, R. Fridström, E. Giovannozzi, F. Imbeaux9, B. Labit, E. de la Luna, C. Maggi, M. Owsiak, R. Scannell y *JET contributors*. «Pedestal structure, stability and scalings in JETILW: the EUROfusion JET-ILW pedestal database», **2021** *Nucl. Fusion* 61 016001. 4.3.1

Anexo: Código

El código desarrollado en el trabajo tiene su fuente en [60], pero ha sido objeto de múltiples modificaciones y mejoras, empezando por la multidimensionalización del mismo. Nuestro código se encuentra alojado en la plataforma GitHub y es de libre acceso, el enlace es https://github.com/alexpanera/Surrogate-Model-EPED-EuroPED. En este repositorio se encuentran 5 archivos:

- El modelo bidimensional: *AP_BNN_NCP.ipynb*.
- El modelo sustituto de EPED: *AP_BNN_EPED_7D.ipynb*.
- El modelo susituto de EuroPED: *AP_BNN_EuroPED_mu.py*.
- El modelo sustituto de EuroPED con β_n como salida añadida: *AP_BNN_EuroPED_mu_beta.ipynb*.
- El modelo sustituto de los datos experimentales: *AP_BNN_exp_4out.ipynb*.

Entre ellos, *AP_BNN_EuroPED_mu.py* es en el que mejor se observan todas las partes de la red neuronal, su entrenamiento, el tratamiento de los datos y las posteriores representaciones.

Sin lugar a duda, desarrollar el código y construir los modelos ha sido la parte más laboriosa del proyecto, dada la complejidad del mismo y la naturaleza de los datos.