



Configuración óptima de una RNA para la resolución del problema cinemático inverso en manipuladores paralelos: caso del 5R

C. Rodríguez¹, A. Hernández¹, O. Altuzarra¹, E. Amezua¹

¹ Dpto. de Ingeniería Mecánica. Universidad del País Vasco UPV/EHU. Escuela de Ingeniería de Bilbao
christopher.rodriguez.afonso@gmail.com, a.hernandez@ehu.eus, oscar.altuzarra@ehu.eus, enrique.amezua@ehu.eus

En este artículo se aborda la optimización y el estudio de hiperparámetros clave (número de neuronas por capa oculta, número de capas ocultas y tipo de funciones de activación) de una RNA (Red Neuronal Artificial), concretamente de un perceptrón multicapa (MLP), para la resolución del problema cinemático inverso (IKP) completo de un manipulador paralelo. Se realiza un estudio en profundidad del efecto de dichos hiperparámetros en el desempeño de la red, tanto a nivel de error en las inferencias a lo largo del espacio de trabajo, como a nivel de coste computacional. Concretamente, se aborda el problema para un manipulador paralelo 5R asimétrico, cuya cinemática es bien conocida, de forma que, dadas una o varias posiciones del elemento terminal, la RNA permita obtener las soluciones del IKP asociadas a todos los modos de trabajo del 5R. Para facilitar las inferencias por parte del usuario final, se introduce la red en una función que incluye un preproceso y postproceso (donde se tiene en cuenta el método empleado para resolver la discontinuidad de los ángulos en el valor de cero radianes) para las variables de entrada y de salida de la red, respectivamente. En este artículo se presentan dos experimentos de variación de hiperparámetros, de los que se extraen una serie de conclusiones. Con estos experimentos se han observado aspectos significativos como la degradación de los resultados al exceder un cierto número de capas ocultas, demostrando la importancia de la selección de unos hiperparámetros adecuados y de no caer en el error de suponer que una red más compleja vaya a otorgar siempre mejores aproximaciones. Este estudio para la resolución del IKP completo de un 5R se puede considerar como un caso particular, pudiendo servir de base para la optimización de hiperparámetros en MLPs que resuelvan el IKP completo para manipuladores paralelos de cinemática más compleja. No obstante, se debe tener presente que los valores óptimos de los hiperparámetros dependen de la complejidad del problema, así como de la cantidad de datos disponibles para el entrenamiento. Para estos últimos manipuladores, los contenidos de este artículo son especialmente beneficiosos, debido al elevado coste computacional de los métodos numéricos empleados para la resolución de su cinemática, frente al coste computacional de las inferencias de una red entrenada.

1. Introducción

Los manipuladores paralelos cuentan con múltiples ventajas con respecto a los manipuladores serie, como, por ejemplo, una mayor rigidez y capacidad de carga, un emplazamiento más conveniente de los actuadores, una mayor precisión y una mayor capacidad para alcanzar velocidades y aceleraciones elevadas [1]. Estas ventajas explican el elevado interés en este tipo de mecanismos. No obstante, los manipuladores paralelos presentan algunas desventajas con respecto a los manipuladores serie, destacando, a nivel general, un espacio de trabajo más reducido y con singularidades internas, y una cinemática más compleja.

Para solventar la reducción generalizada del espacio de trabajo de los manipuladores paralelos sin renunciar a las ventajas inherentes de éstos, existen ciertos tipos de manipuladores paralelos que permiten alcanzar grandes espacios de trabajo, como, por ejemplo, los manipuladores paralelos actuados por cables (más conocidos por CDPRs, *Cable-Driven Parallel Robots*). Sin embargo, la mayor complejidad cinemática de los manipuladores paralelos sigue presente (más aún en los CDPRs y otros manipuladores paralelos de elementos flexibles), careciendo la mayoría de ellos de solución analítica para los problemas cinemáticos directo e inverso (DKP e IKP, respectivamente). Ante esta ausencia de solución analítica, se suele recurrir a métodos numéricos computacionalmente costosos.

Este hecho explica el interés en el uso de RNAs (Redes Neuronales Artificiales) para la resolución de los problemas cinemáticos en manipuladores paralelos [2-3]. En algunos casos se emplean métodos híbridos que combinan redes neuronales con métodos numéricos, de forma que las inferencias de las redes se emplean como primera aproximación para los métodos numéricos, reduciendo el número de iteraciones necesarias para los mismos [4-6].

La reducción en el coste computacional de la resolución de los problemas cinemáticos es de interés tanto para la etapa de síntesis del manipulador paralelo (durante la cual se requieren cuantiosas resoluciones de dichos problemas) como para su control en tiempo real, ya que estos manipuladores suelen emplearse para movimientos más rápidos que los alcanzables con manipuladores serie.

En este artículo se aborda la optimización y el estudio de algunos de los hiperparámetros más importantes de un perceptrón multicapa (MLP), que es un tipo de RNA, para la resolución del IKP de un manipulador paralelo, obteniendo las soluciones asociadas a todos los modos de trabajo del mismo. Concretamente, los hiperparámetros analizados en este artículo son el número de neuronas por capa oculta, el número de capas ocultas y el tipo de funciones de activación (probadas estas funciones con diferentes métodos de inicialización de los pesos del MLP). El impacto de estos hiperparámetros ha sido estudiado tanto a nivel de coste computacional como a nivel de precisión de las inferencias del MLP, y para dicho estudio ello se han diseñado dos experimentos factoriales.

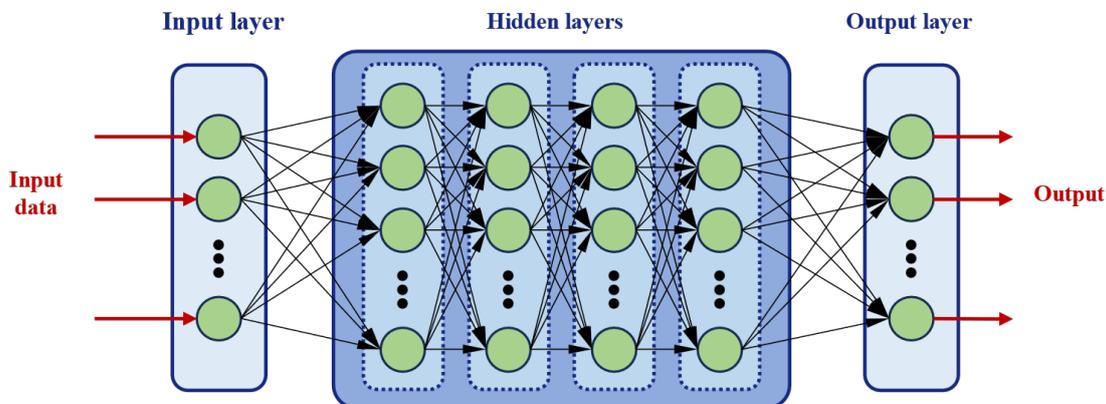


Figura 1: Esquema de una red neuronal MLP.

El estudio se ha realizado concretamente para la resolución del IKP de un manipulador paralelo plano 5R. A pesar de que éste es un manipulador paralelo cuyo IKP puede resolverse de forma analítica, por lo que no es computacionalmente costoso, el estudio realizado sobre este mecanismo puede considerarse como una particularización de una metodología más general, pudiendo los resultados aquí obtenidos servir de base para la optimización de los hiperparámetros de otros MLPs orientados a la resolución del IKP en mecanismos de cinemática más compleja. De hecho, algunas de las conclusiones obtenidas en este artículo (como la observación de la degradación de resultados ante un incremento excesivo del número de capas ocultas) son directamente extrapolables a la resolución del IKP en otros mecanismos mediante el empleo de MLPs.

Para aplicar esta metodología a manipuladores paralelos más complejos se deben tener en cuenta diversos factores. Por una parte, los manipuladores paralelos de cinemática más compleja que la del 5R también presentan singularidades del IKP más complejas. Siendo en las inmediaciones de dichas singularidades donde se esperan los mayores errores de inferencia (puesto que estas singularidades marcan los límites del espacio de trabajo). Por tanto, para obtener el mismo error que en el 5R, es de esperar que se requieran más puntos en el set de entrenamiento

cercanos a estas singularidades, para así poder caracterizar adecuadamente la mayor complejidad geométrica del espacio de trabajo.

Otro aspecto a tener en cuenta es la dimensionalidad del problema: al aumentar el número de entradas y salidas, aumenta la dimensionalidad de la función a aproximar por la red neuronal. Asimismo, los manipuladores de cinemática más compleja también presentan relaciones no lineales más complejas entre sus entradas y salidas. Todo esto, unido a una geometría más compleja del espacio de trabajo, hace que aumente la complejidad de la función a aproximar por la red, pudiendo requerirse de redes neuronales más anchas (más neuronas por capa oculta) y/o profundas (más capas ocultas). Sin embargo, el aumento de parámetros de la red tiende a disminuir su capacidad de generalización; pudiendo hacer necesario el empleo de técnicas de regularización. Asimismo, las redes más profundas son más difíciles de entrenar, pudiendo no ser suficiente con el empleo de técnicas como batch normalization [7], pasando a necesitar de modificaciones en la estructura de la red, como la implementación de conexiones de salto (como, por ejemplo, las que se emplean en las redes neuronales residuales [8]).

2. Resolución cinemática del manipulador paralelo 5R

El 5R analizado en este artículo es un 5R asimétrico, cuyo diagrama cinemático y dimensiones principales se presentan en la Figura 2, siendo estas dimensiones principales las mismas que las del 5R analizado en [1]. Éste se trata de un manipulador paralelo plano de 2 grados de libertad (gdl), siendo sus coordenadas generalizadas los ángulos θ_1 y θ_2 (Figura 2), y siendo las coordenadas de P (x_p e y_p), las variables de salida del mecanismo.

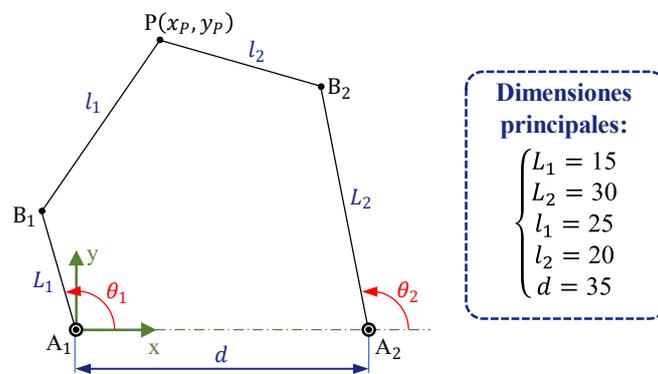


Figura 2: Diagrama cinemático y dimensiones principales del 5R analizado.

La cinemática del 5R ha sido objeto de estudio por múltiples investigadores, lo que facilita posibles futuras ampliaciones del trabajo llevado a cabo en este artículo a otras cuestiones cinemáticas del 5R.

En este documento se ha empleado la misma notación que en [1] para designar a las configuraciones de las cadenas cinemáticas del 5R, a los modos de trabajo del mismo y a sus modos de ensamblado.

2.1. Problema cinemático inverso (IKP) del 5R

El IKP consiste en obtener el valor de θ_1 y θ_2 , teniendo como dato las coordenadas x_p e y_p del punto P.

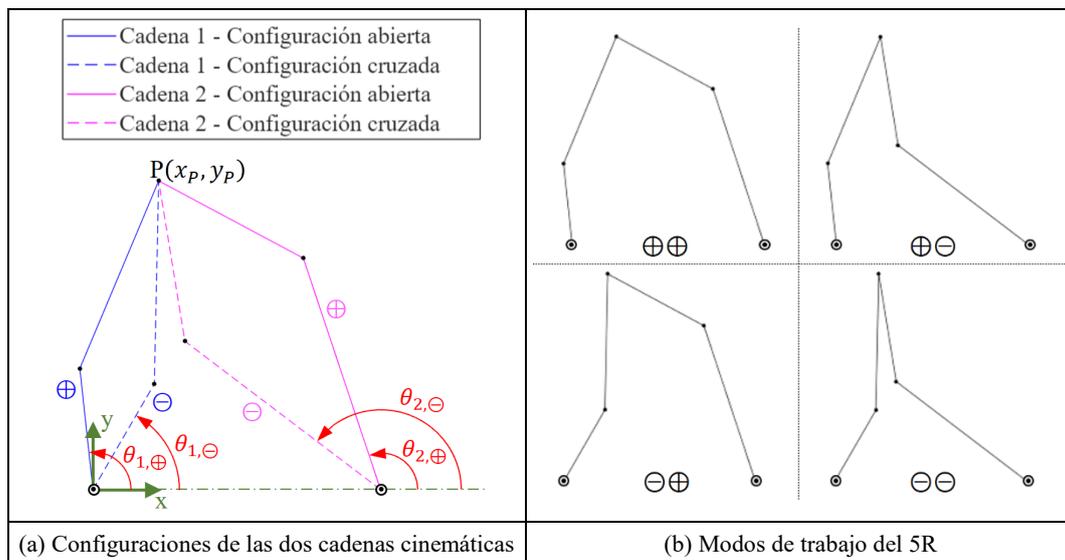


Figura 3: Soluciones del IKP del 5R.

Para una posición del punto P dada, salvo que sea una posición de singularidad del problema inverso (SPI), las dos cadenas cinemáticas pueden adquirir dos posibles configuraciones -Figura 3 (a)-. De esta forma, las coordenadas generalizadas pueden tomar dos posibles valores cada una ($\theta_{i,\oplus}, \theta_{i,\ominus} / i \in \{1,2\}$). Las 4 posibles combinaciones de los θ_i dan lugar a 4 soluciones del IKP del 5R, que constituyen los 4 modos de trabajo del 5R -Figura 3 (b)-.

Las SPI del 5R están caracterizadas por el alineamiento de, al menos, uno de los pares de barras L_i y l_i . En dichas posiciones se tiene que $\theta_{i,\oplus} = \theta_{i,\ominus}$, siendo i el par de barras alineadas -Figura 4 (a)-. Asimismo, las SPI definen los límites del espacio de trabajo total del mecanismo.

A continuación, se expone un método de resolución analítico del IKP del 5R, apoyando las explicaciones sobre el diagrama cinemático de la Figura 4 (b).

Las longitudes h_1 y h_2 , expresadas en función de x_p e y_p , son:

$$\begin{cases} h_1 = \sqrt{x_p^2 + y_p^2} \\ h_2 = \sqrt{(d - x_p)^2 + y_p^2} \end{cases} \quad (1)$$

Tras ello, se aplica el teorema del coseno a los dos triángulos resaltados en la Figura 4 (b):

$$l_i^2 = L_i^2 + h_i^2 - 2 \cdot L_i \cdot h_i \cdot \cos(\gamma_i) \quad \forall i \in \{1,2\} \Rightarrow \cos(\gamma_i) = \frac{L_i^2 + h_i^2 - l_i^2}{2 \cdot L_i \cdot h_i} \quad \forall i \in \{1,2\} \quad (2)$$

$$\begin{cases} \gamma_{i,1} = \arccos\left(\frac{L_i^2 + h_i^2 - l_i^2}{2 \cdot L_i \cdot h_i}\right) \\ \gamma_{i,2} = 2\pi - \arccos\left(\frac{L_i^2 + h_i^2 - l_i^2}{2 \cdot L_i \cdot h_i}\right) \end{cases}$$

De las dos posibles soluciones de γ_i , se coge únicamente la de menor ángulo, con lo que se tiene que:

$$\gamma_i = \arccos\left(\frac{L_i^2 + h_i^2 - l_i^2}{2 \cdot L_i \cdot h_i}\right) \quad \forall i \in \{1,2\} \quad (3)$$

Por otra parte, los ángulos ψ_1 y ψ_2 son:

$$\begin{cases} \psi_1 = \text{atan2}(y_p, x_p) \\ \psi_2 = \text{atan2}(y_p, x_p - d) \end{cases} \quad (4)$$

Finalmente, los valores de los cuatro distintos θ_i , soluciones del IKP del 5R, son:

$$\begin{cases} \theta_{1,\oplus} = \psi_1 + \gamma_1 \\ \theta_{1,\ominus} = \psi_1 - \gamma_1 \\ \theta_{2,\oplus} = \psi_2 - \gamma_2 \\ \theta_{2,\ominus} = \psi_2 + \gamma_2 \end{cases} \quad (5)$$

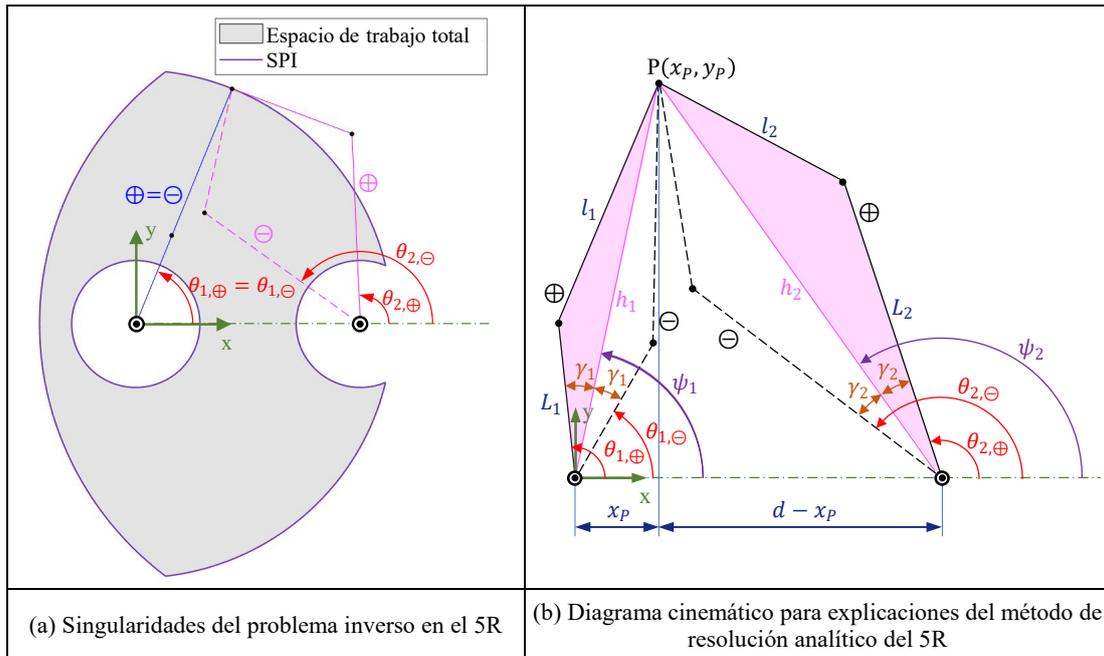


Figura 4: Aspectos varios del IKP del 5R.

2.2. Problema cinemático directo (DKP) del 5R

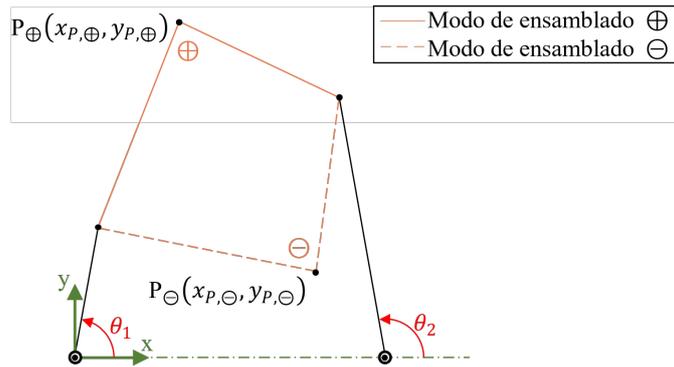


Figura 5: Soluciones del DKP del 5R.

Aunque en este artículo los MLPs se han empleado únicamente para resolver el IKP del 5R, es relevante hablar de las singularidades del problema directo (SPD) del 5R, ya que definen algunos de los límites de los espacios de trabajo asociados a las distintas combinaciones de modos de trabajo y modos de ensamblado del mecanismo.

El DKP consiste en obtener el valor de las coordenadas x_p e y_p , teniendo como dato el valor de θ_1 y θ_2 . Salvo en SPD, el DKP del 5R presenta 2 soluciones, es decir, 2 posibles posiciones del punto P (Figura 5). Estas 2 soluciones constituyen los 2 modos de ensamblado del 5R.

En el 5R las SPD están caracterizadas por el alineamiento de las barras l_1 y l_2 . Asimismo, cabe destacar que existe una asociación entre las SPD y los distintos modos de trabajo del mecanismo (Figura 6).

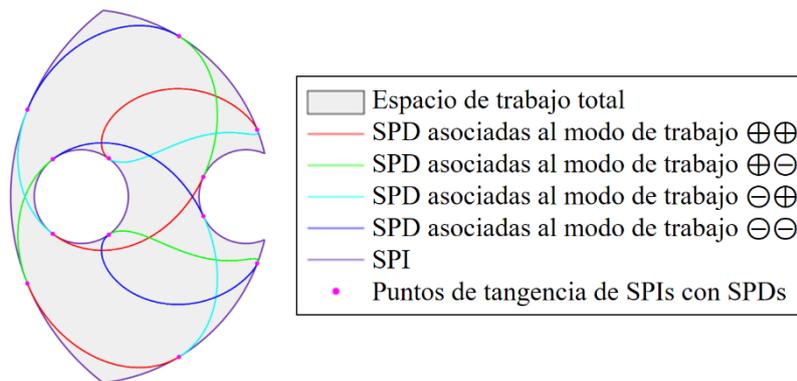


Figura 6: SPD del 5R asociadas a los distintos modos de trabajo del mismo, representadas junto con el espacio de trabajo total del 5R y sus SPI.

Para cambiar de modo de ensamblado del 5R sin desmontarlo es necesario pasar por una SPD (no aconsejable ya que en ellas se produce una pérdida del control del mecanismo [1]).

Los límites de los espacios de trabajo asociados al 5R en sus distintos modos de trabajo y de ensamblado (Figura 7) vienen dados por la combinación de las SPI junto con las SPD correspondientes. Estos límites se han considerado al confeccionar el set de entrenamiento para los MLPs. Cerca de estos límites se ha incrementado la densidad de puntos del set de entrenamiento de cara a evitar grandes errores de los MLPs en dichas zonas.

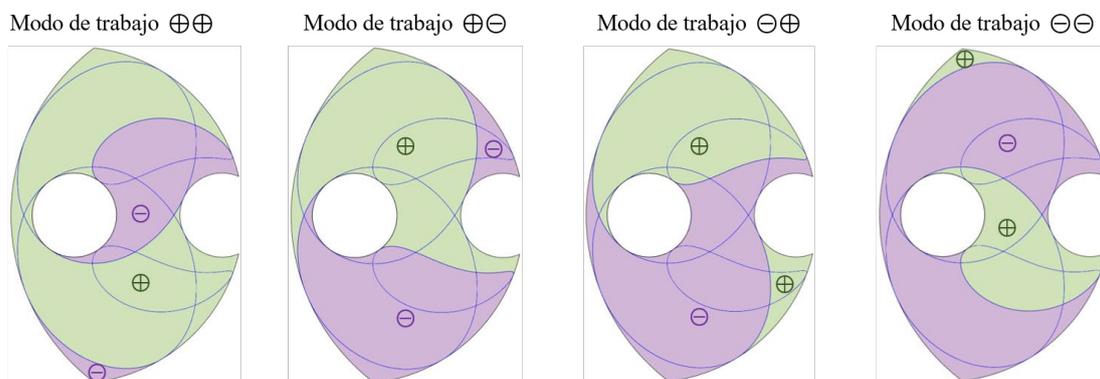


Figura 7: Espacios de trabajo del 5R asociados a los distintos modos de trabajo y modos de ensamblado.

3. Resolución del IKP del 5R haciendo uso de MLPs

3.1. Selección de variables de entrada y salida de los MLPs

Una posible forma de resolver el IKP del 5R para todos sus modos de trabajo es el empleo de un MLP que tenga como variables de entrada las coordenadas x_P e y_P del punto P, y como variables de salida, los ángulos $\theta_{1,\oplus}$, $\theta_{1,\ominus}$, $\theta_{2,\oplus}$ y $\theta_{2,\ominus}$ (soluciones del IKP).

Sin embargo, este planteamiento tiene un problema: las variables angulares presentan una discontinuidad en el paso de 2π a 0 radianes (Figura 8). Teniendo en cuenta que las funciones de activación usadas en MLPs suelen ser continuas (de lo contrario, no podrían entrenarse mediante algoritmos de retropropagación, backpropagation en inglés, aunque sí con algoritmos genéticos), una discontinuidad en las variables de salida implica varios problemas:

- Imposibilitaría alcanzar un valor nulo de la función de coste si alguna de las variables de salida tuviese un valor teórico (valor objetivo) de 0 radianes, puesto que para la correspondiente entrada se tendrían dos posibles valores de salida: 0 radianes y 2π radianes. Esto dificultaría el proceso de entrenamiento del MLP, durante el cual se trata de minimizar el valor de la función de coste para el set de entrenamiento.
- Tras el entrenamiento, se obtienen inferencias de error elevado para ángulos próximos a 0 radianes.

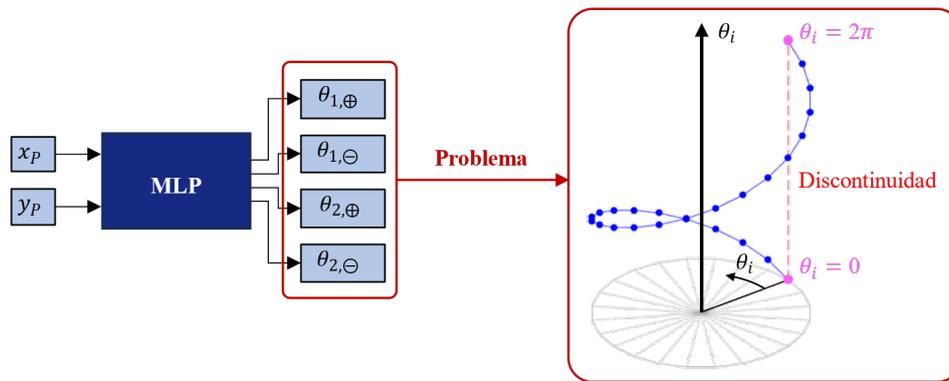


Figura 8: Problema si se empleasen directamente las coordenadas generalizadas del 5R como variables de salida de un MLP.

Para solventar este problema se ha decidido emplear como variables de salida del MLP los senos y cosenos de los ángulos θ_i del 5R. De forma que, en vez de 4 variables de salida discontinuas, el MLP cuenta con 8 variables de salida continuas. Los ángulos θ_i correspondientes a estos senos y cosenos se pueden obtener luego, fácilmente, mediante funciones arcotangente de doble argumento (Figura 9).

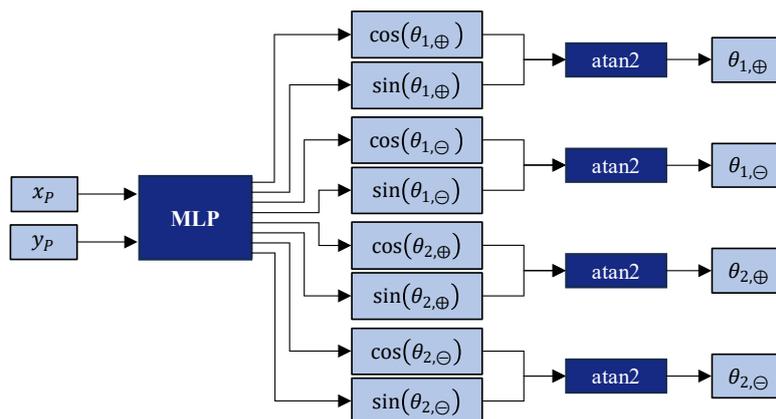


Figura 9: MLP para resolución del IKP del 5R.

3.2. Preproceso de variables de entrada y objetivos de salida de los MLPs y postproceso de inferencias

Para distinguir entre las variables de salida del MLP y los valores teóricos de las mismas, contenidos en los sets (de entrenamiento validación y test) y asociados a unas entradas concretas, a estos valores teóricos se les denominará, de ahora en adelante, objetivos de salida del MLP.

Otro término importante es el de inferencia, que puede referenciar tanto a las predicciones que otorga el MLP para las variables de salida ante unos valores de las variables de entrada (inferencia entendida como predicción); como al proceso que tiene lugar en el MLP para la obtención de dichas predicciones (inferencia entendida como proceso).

El preproceso de las variables de entrada y objetivos de salida de una red se entiende como aquellas transformaciones aplicadas a dichas variables antes de conformar con ellas los distintos sets. Este preproceso es, normalmente, una buena práctica al trabajar con redes neuronales, ya que, entre otros aspectos, suele ayudar a una convergencia más rápida de la función de coste hacia valores mínimos (ejemplo en la Figura 10 para un MLP ficticio con dos parámetros a optimizar: w_1 y w_2).

Para todos los sets empleados en este artículo se ha optado por emplear la normalización min-max en el rango $[-1, 1]$ tanto para las variables de entrada como para los objetivos de salida:

$$\{v'\} = \frac{\{v\} - \{1\} \cdot v_{min}}{v_{max} - v_{min}} \cdot (u - l) + \{1\} \cdot l \quad (6)$$

Donde $\{v\}$ es el vector a normalizar, $\{v'\}$ es el vector normalizado, u y l son los límites superior e inferior, respectivamente, del rango al cual se quiere normalizar, v_{max} y v_{min} son los elementos con valores máximo y mínimo, respectivamente, del vector $\{v\}$, y $\{1\}$ tiene la misma dimensión que $\{v\}$.

Las variables de salida de los MLPs son senos y cosenos, de por sí ya limitados al rango $[-1, 1]$. El sentido detrás de su normalización es que, por las dimensiones principales del 5R, el ángulo $\theta_{2,\Theta}$ no da vueltas completas, por lo que sus senos y cosenos están limitados a un rango menor que el resto de senos y cosenos.

La expresión (6) se aplica a cada una de las variables a normalizar del set de entrenamiento. Para normalizar las variables del resto de sets (validación y test), se aplica también la expresión (6), pero, en vez de extraer los máximos y los mínimos de dichos sets, se emplean los máximos y mínimos que se extrajeron durante la normalización de las variables en el set de entrenamiento, ya que el preproceso debe ser el mismo en todos los sets.

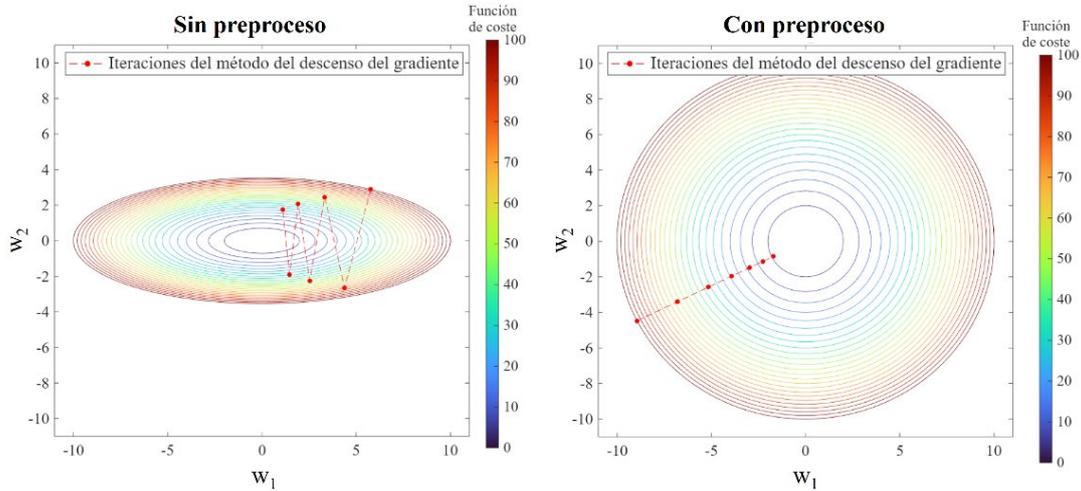


Figura 10: Efecto del preproceso de las variables en la función de coste.

Esta normalización implica que, para que el usuario final pueda realizar inferencias con el MLP, es necesario incluir este MLP dentro de una función como la representada en la Figura 11. Esto se ha tenido en cuenta en las evaluaciones del coste computacional.

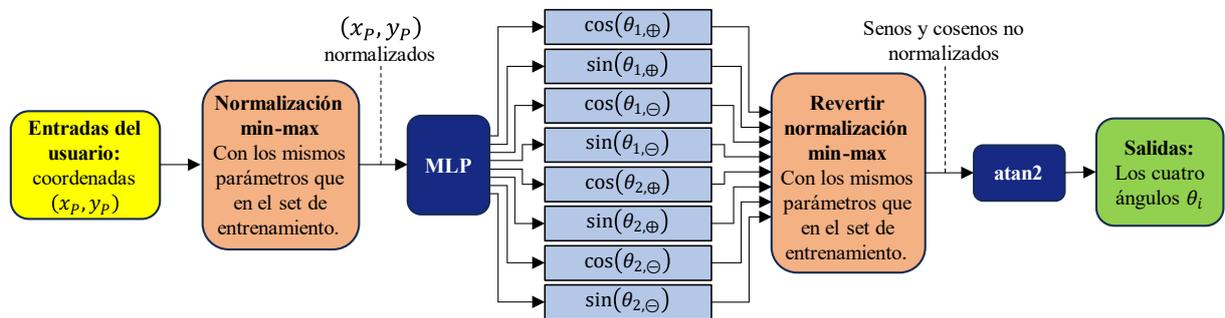


Figura 11: Función para resolver el IKP del 5R haciendo uso de un MLP.

El postproceso se realiza únicamente sobre las inferencias de la red, y consiste en la reversión de la normalización min-max, así como en la aplicación de la función atan2 a los senos y cosenos no normalizados (Figura 11).

3.3. Aspectos comunes a todos los MLPs estudiados en este artículo

Los siguientes aspectos o características son comunes a todos los MLPs analizados en los dos experimentos de este artículo:

- Son redes de regresión, con la función identidad como función de activación de la capa de salida.
- El número de neuronas y las funciones de activación se mantienen constantes para todas las capas ocultas.
- Batch normalization [7] aplicada a las activaciones de todas las capas ocultas.
- Al comienzo del entrenamiento todos los sesgos se han inicializado a un valor nulo.
- Se ha empleado Adam [9] como optimizador.
- Se ha empleado el MSE como función de coste para el entrenamiento de los MLPs.
- El entrenamiento se ha llevado a cabo por batches en lugar de por mini-batches, es decir, cada iteración del proceso de entrenamiento supone un epoch.
- La función de coste no incluye ningún término de regularización (típicos de técnicas de regularización de pesos como las regularizaciones L1 y L2 o la Elastic Net [10]), ya que en pruebas preliminares no se detectaron problemas de sobreajuste (más conocido por el término inglés de overfitting).
- El único criterio de parada establecido es un límite 100,000 epochs, que es un número elevado teniendo en cuenta la relativa simplicidad del problema, pero así se asegura que los MLPs con elevado número de parámetros (los cuales tienden a requerir una mayor cantidad de epochs para su entrenamiento) sean comparados con otros MLPs en igualdad de condiciones.
- Se comienza el entrenamiento con una tasa de aprendizaje de valor 0.0015, la cual se multiplica por un valor de 0.2 cada 20,000 epochs. De nuevo, es un número de epochs elevado, pero de esta forma se asegura que todas las redes hayan sido entrenadas durante un tiempo suficientemente largo con cada uno de los distintos valores tomados por la tasa de aprendizaje.
- Se ha evaluado la función de coste para el set de validación cada 25 epochs.
- El valor de los parámetros proporcionado tras el entrenamiento es el correspondiente a la evaluación que haya arrojado un menor valor de la función de coste en el set de validación. De esta forma se evita que los resultados estén condicionados por el posible overfitting debido al elevado número de epochs.
- Todos los entrenamientos se han llevado a cabo usando el software MATLAB®.

3.4. Sets empleados en los experimentos

Se han empleado 3 tipos de sets: de entrenamiento, de validación y de test. Todos los sets se han mantenido constantes a lo largo de los dos experimentos.

El set de entrenamiento se ha confeccionado distribuyendo los puntos según una cuadrícula en el espacio de trabajo total del 5R, pero añadiendo puntos adicionales en las zonas alrededor de las SPI y SPD. En total, el set de entrenamiento cuenta con 4818 puntos.

Por su parte, los puntos del set de validación han sido escogidos con un método pseudoaleatorio dentro del espacio de trabajo total del 5R. En total el set de validación cuenta con 964 puntos.

Por último, se han empleado 4 sets de test distintos, con el objetivo de extraer algunas conclusiones de interés mediante la comparación del desempeño de los MLPs en dichos sets.

Tabla 1: Características de los 4 sets de test empleados.

	Características del set	
	Número de puntos	Criterio de selección de los puntos
Set de Test 1	643	Puntos seleccionados de forma pseudoaleatoria
Set de Test 2	633	Puntos distribuidos según cuadrícula rectangular en plano (x_p, y_p)
Set de Test 3	15167	Puntos seleccionados de forma pseudoaleatoria
Set de Test 4	15167	Puntos distribuidos según cuadrícula rectangular en plano (x_p, y_p)

Los sets de test 1 y 2 representan casos con proporciones típicas (75% entrenamiento, 15% validación y 10% test), mientras que los sets de test 3 y 4 representan casos poco prácticos, con muchos más puntos en los sets de test que en los sets de entrenamiento y validación (Tabla 1).

4. Diseño de los dos experimentos para optimización de hiperparámetros

Para el estudio de hiperparámetros llevado a cabo en este artículo, se han diseñado dos experimentos factoriales.

4.1. Experimento 1

En este experimento se estudia el efecto del número de neuronas por capa oculta y del número de capas ocultas.

Téngase en cuenta que, en este artículo, cuando se haga mención al número de capas ocultas, se estarán contando únicamente las capas completamente conectadas, excluyendo del conteo sus respectivas capas de activación y de batch normalization. Con esto presente, los niveles definidos para los factores tratamiento del experimento 1 son:

- Número de neuronas por capa oculta (14 niveles): {25, 50, 75, 100, 125, 150, 175, 200, 250, 300, 350, 400, 450, 500}.
- Número de capas ocultas (11 niveles): {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}.

Por tanto, es un experimento factorial 14×11 , dando un total de 154 tratamientos o arquitecturas distintas de MLPs. Además, cada arquitectura ha sido entrenada 3 veces (3 repeticiones por arquitectura) para reducir la dependencia de las conclusiones con respecto a los valores de inicialización de los pesos. Por lo que para el experimento 1 se han entrenado un total de 462 MLPs, a lo largo de los cuales, además de los aspectos mencionados en el subapartado 3.3, se mantienen constantes los siguientes hiperparámetros:

- Se emplea la función de activación PReLU [11] en todas las capas ocultas.
- Los pesos se inicializan con la inicialización de He [11].

4.2. Experimento 2

Empleando el número de neuronas por capa oculta y el número de capas ocultas seleccionado como óptimo del experimento 1, en este experimento se estudia el rendimiento para la resolución del IKP del 5R de algunas de las funciones de activación más populares en redes neuronales.

Cada una de estas funciones de activación se prueba con 3 algoritmos de inicialización de pesos distintos, para no condicionar las conclusiones al hecho de que algunas funciones de activación puedan rendir mejor con un tipo de inicialización determinado.

De esta forma, los factores tratamiento del experimento 2 son la función de activación de las capas ocultas y el algoritmo de inicialización de los pesos. Los niveles definidos para cada uno de estos factores tratamiento son:

- Función de activación de las capas ocultas (6 niveles): {tanh, sigmoide, ReLU, PReLU [11], Swish [12] (con $\beta = 1$, de forma que es equivalente a la función SiLU [13]), Mish [14]}.
- Algoritmo de inicialización de los pesos (3 niveles): {inicialización de Glorot [15], inicialización de He [11], inicialización Narrow-Normal de MATLAB® (realizando un muestreo independiente de una distribución normal con media cero y desviación estándar de 0.01)}

Se trata de un experimento factorial 6×3 , dando un total de 18 tratamientos, cada uno de los cuales ha sido repetido 7 veces para extraer conclusiones robustas, dando un total de 126 MLPs entrenados para el experimento 2.

5. Hardware empleado

Para el estudio llevado a cabo en este artículo se ha empleado un ordenador con las siguientes características:

- CPU: Intel® Core™ i9-14900HX
- GPU: NVIDIA® GeForce RTX™ 4070 Laptop
- RAM: 32 GB DDR5 5600 MT/s

6. Resultados del experimento 1

6.1. Efectos en el coste computacional

Para cada uno de los MLPs entrenados en el experimento 1 se ha cronometrado la resolución del IKP del 5R (usando una función como la representada en la Figura 11) para los 15167 puntos del set de test 4.

Como los tiempos de resolución son relativamente bajos, éstos pueden verse afectados por procesos en segundo plano. Así que, se han realizado un número de mediciones lo suficientemente elevado para garantizar la obtención de tiempos representativos. Una vez obtenido el tiempo de resolución para los 462 MLPs, se ha obtenido el tiempo de resolución promedio para cada una de las 154 arquitecturas de MLPs estudiadas en el experimento 1.

En cuanto a los tiempos de las diferentes operaciones representadas en la Figura 11 se debe tener presente que el tiempo para preprocesar las entradas proporcionadas por el usuario, así como el tiempo para postprocesar las inferencias del MLP son independientes del número de capas ocultas y del número de neuronas por capa oculta. Por tanto, se puede concluir que el efecto de dichas operaciones sobre las curvas de la Figura 12 y la Figura 13 se limita a un desplazamiento de las mismas en ordenadas, no afectando ni a la curvatura ni a la pendiente.

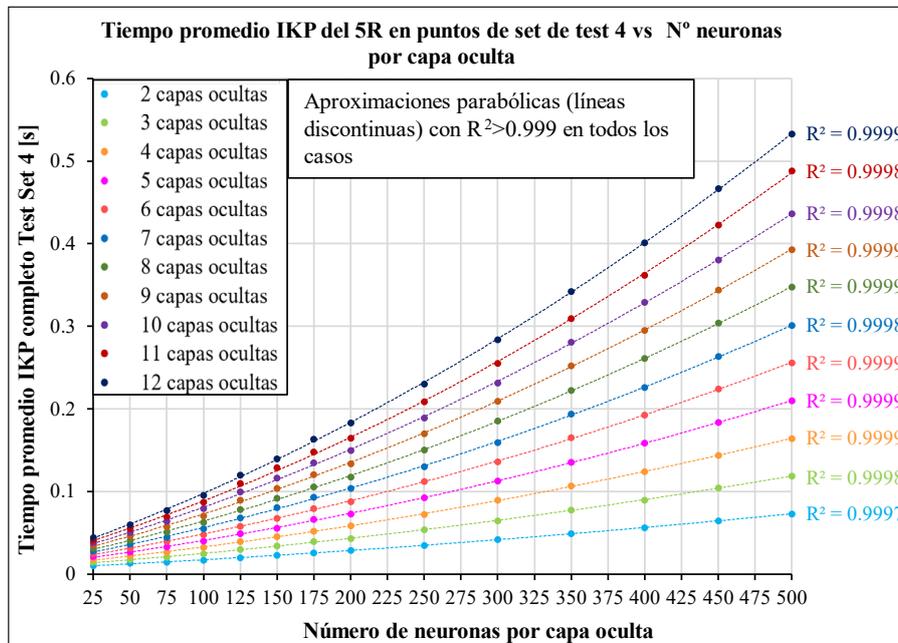


Figura 12: Tiempo promedio de resolución del IKP completo del 5R para todos los puntos del set de test 4. Evolución del coste computacional con respecto del número de neuronas por capa oculta.

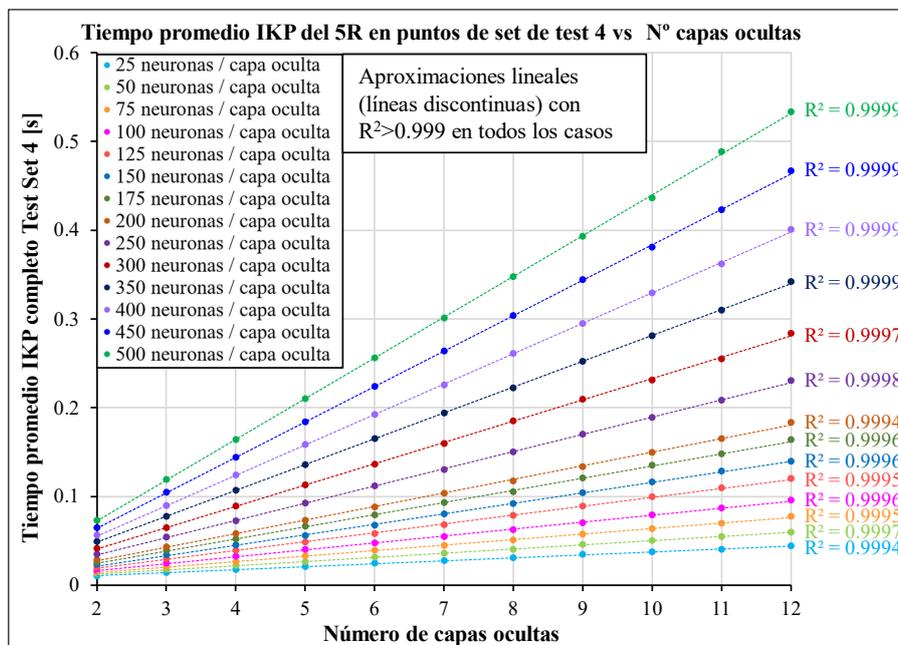


Figura 13: Tiempo promedio de resolución del IKP completo del 5R para todos los puntos del set de test 4. Evolución del coste computacional con respecto del número de capas ocultas.

En el caso de la Figura 12, los datos (agrupados según el número de capas ocultas) se aproximan muy bien usando curvas polinómicas de segundo orden, ya que se obtienen valores del coeficiente de determinación, R^2 , muy cercanos a 1 ($R^2 > 0.999$ en todos los casos estudiados). Visualmente se puede apreciar la curvatura de los resultados, por lo que una aproximación lineal no sería adecuada para aproximar los resultados de la Figura 12.

En el caso de la Figura 13 una aproximación lineal de los datos (agrupados según el número de neuronas por capa oculta) sí que resulta adecuada, obteniéndose $R^2 > 0.999$ en todos los casos estudiados.

De esta forma se puede concluir lo siguiente, al menos dentro los rangos estudiados de número de capas ocultas y número de neuronas por capa oculta, para MLPs con funciones de activación PReLU, con batch normalization aplicada a las activaciones de cada capa oculta y con un número de neuronas constante a lo largo de sus capas ocultas: el coste computacional crece de forma parabólica con el número de neuronas por capa oculta (lo que indica que el número de conexiones entre capas ocultas tiene un peso relativamente elevado dentro del coste computacional de las inferencias de estos MLPs) y crece linealmente con el número de capas ocultas.

6.2. Efectos en el error de las inferencias

Se ha calculado el MSE en los distintos sets (set de entrenamiento, set de validación y los 4 sets de test) para cada uno de los 462 MLPs entrenados en el experimento 1. De dichos resultados se han extraído el MSE promedio y el MSE mínimo de entre las 3 repeticiones para cada una de las 154 arquitecturas de MLPs.

En primer lugar, se va a justificar el hecho de que el MSE en los sets de test 3 o 4 es más representativo del error de los MLPs a lo largo del espacio de trabajo del 5R que el MSE en los sets de test 1 o 2. El hecho de trabajar con sets de test tan grandes como el 3 y el 4 (grandes en comparación a los sets de entrenamiento y validación) no es práctico ni aconsejable en la mayoría de ocasiones. No obstante, la justificación antes mencionada permitirá a su vez justificar otra serie de conclusiones de mayor relevancia y que se recogen más adelante en este artículo.

En la Figura 14 se puede apreciar que la influencia del método de selección de los puntos -de forma pseudoaleatoria o mediante una cuadrícula rectangular en el plano (x_p, y_p) - disminuye al aumentar el tamaño de los sets de test. Esto era esperable, puesto que a medida que se aumenta el número de puntos, el MSE en sets con puntos obtenidos de ambas formas tiende más hacia el MSE del MLP en la totalidad del espacio de trabajo del 5R.

Con esto se puede concluir que el MSE en los sets de test 3 o 4 es más representativo del error de los MLPs a lo largo del espacio de trabajo del 5R que el MSE en los sets de test 1 o 2. Más adelante se verá también que los sets de test 3 y 4 dan lugar a superficies del MSE en función del número de neuronas por capa oculta y del número de capas ocultas similares a la obtenida con el set de validación.

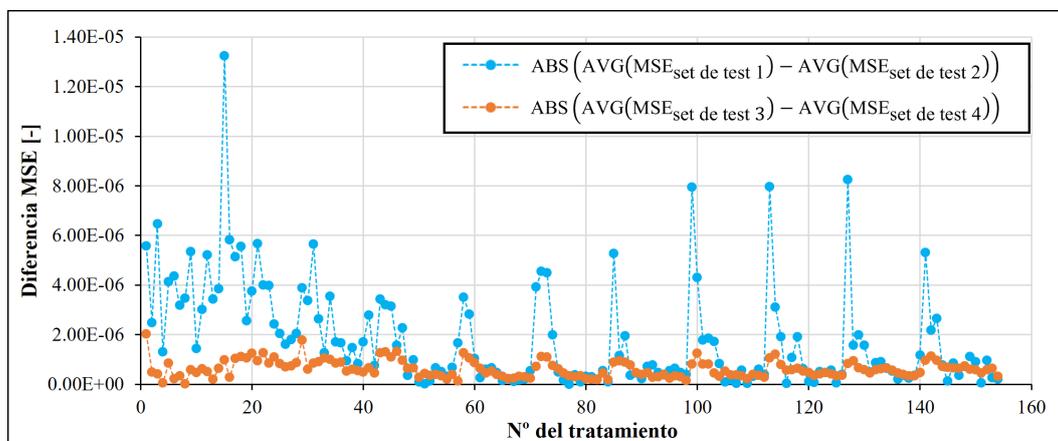


Figura 14: Efecto de la forma en la que se seleccionan los puntos de los sets de test. Evaluación sobre los MLPs del experimento 1.

En la Figura 15 (a) se puede apreciar que, a nivel general, a medida que se aumenta el número de neuronas por capa oculta se reduce el MSE (al menos hasta el límite de 500 neuronas por capa oculta del experimento 1). También se observa que las reducciones absolutas del MSE con el aumento del número de neuronas por capa oculta se ven reducidas a medida que aumenta dicho número de neuronas. De esta forma, se observa que las curvas del MSE promedio de la Figura 15 (a) acaban tendiendo a una asíntota horizontal a medida que se incrementa el número de neuronas por capa oculta.

Se observa que el valor del MSE hacia el que se tiende en el caso de 2 capas ocultas es sustancialmente mayor que el valor del MSE al que se tiende en el caso de 3 capas ocultas, el cual es, a su vez, mayor que el valor al que se tiende en el caso de 4 capas ocultas -zoom de la Figura 15 (a)-. Entre 4 capas ocultas y 12 capas las diferencias en el valor MSE hacia el que se tiende son mucho menos acusadas.

No obstante, se ha de tener en cuenta que si se hubiese empleado un set de entrenamiento de menor tamaño es posible que los MLPs empezasen a sufrir problemas acusados de overfitting para números de neuronas por capa oculta elevados. Estos problemas de overfitting podrían causar que los valores del MSE en el set de validación empeorasen a partir de un cierto número de neuronas por capa oculta, por mucho que de los entrenamientos se extraigan los parámetros para los que se obtiene un menor valor de la función de coste en el set de validación. Por este motivo, no se puede concluir que el hecho de incrementar el número de neuronas por capa oculta siempre vaya a suponer una disminución en el MSE de las inferencias de los MLPs. En la Figura 15 (b) se aprecia que cuando se parte de 2 capas ocultas, se disminuye en gran medida el MSE al añadir una capa oculta adicional, especialmente en los casos con un menor número de neuronas por capa oculta. Asimismo, a medida que se aumentan el número de capas ocultas se observa que el beneficio obtenido por cada capa oculta adicional decae.

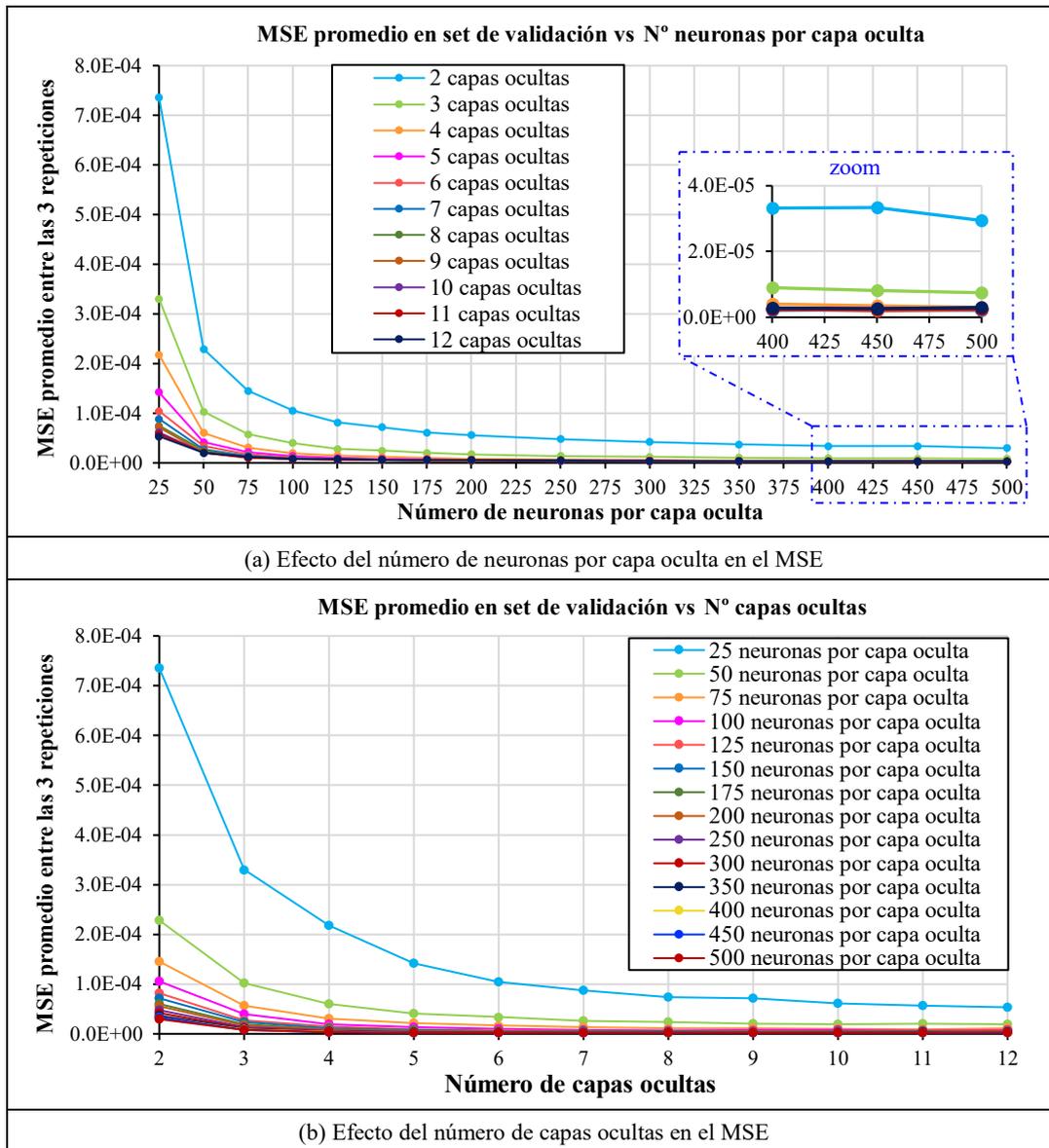


Figura 15: MSE promedio de las inferencias de los MLPs del experimento 1 en el set de validación. Evolución con el número de neuronas por capa oculta y con el número de capas ocultas.

Para gran parte de los MLPs del experimento 1 se ha observado un aumento del MSE en el set de validación al exceder un cierto número de capas ocultas -Figura 16 (a)-. El caso para el que más claramente se aprecia este aumento del MSE es el de 500 neuronas por capa oculta -Figura 16 (b)-. Para determinar si esta tendencia al alza del MSE se trata de un problema de degradación con el número de capas ocultas o de un simple problema de overfitting, se ha de analizar la evolución del MSE en los sets de entrenamiento y validación durante el proceso de entrenamiento, comparando dos arquitecturas de MLPs entre las que se aprecie el aumento del MSE con el aumento número de capas ocultas.

Se han comparado las arquitecturas de 500 neuronas por capa oculta de 8 y 12 capas ocultas (arquitecturas 98 y 154, respectivamente, de la Figura 17). Para ambas se han extraído los históricos del MSE promediados entre las 3 repeticiones en ambos sets (entrenamiento y validación). Posteriormente, a estos históricos promediados se les ha aplicado una media exponencial móvil (de la forma $y_n = \beta \cdot y_{n-1} + (1 - \beta) \cdot x_n$) para suavizar las curvas.

En la Figura 17 se aprecia un incremento del MSE al pasar de la arquitectura 98 (8 capas ocultas) a la 154 (12 capas ocultas) no sólo para el set de validación, sino también para el set de entrenamiento, de forma que se puede concluir que se tiene degradación con el aumento del número de capas ocultas.

No obstante, el empeoramiento de los resultados en el set de validación también se debe a una menor capacidad de generalización debido al aumento del número de parámetros de la red, ya que en el tramo de 50000 a 60000 epochs, se ve que ambas arquitecturas presentan un MSE parecido para el set de entrenamiento, pero con un MSE en el set de validación mayor para la arquitectura 154 que para la 98.

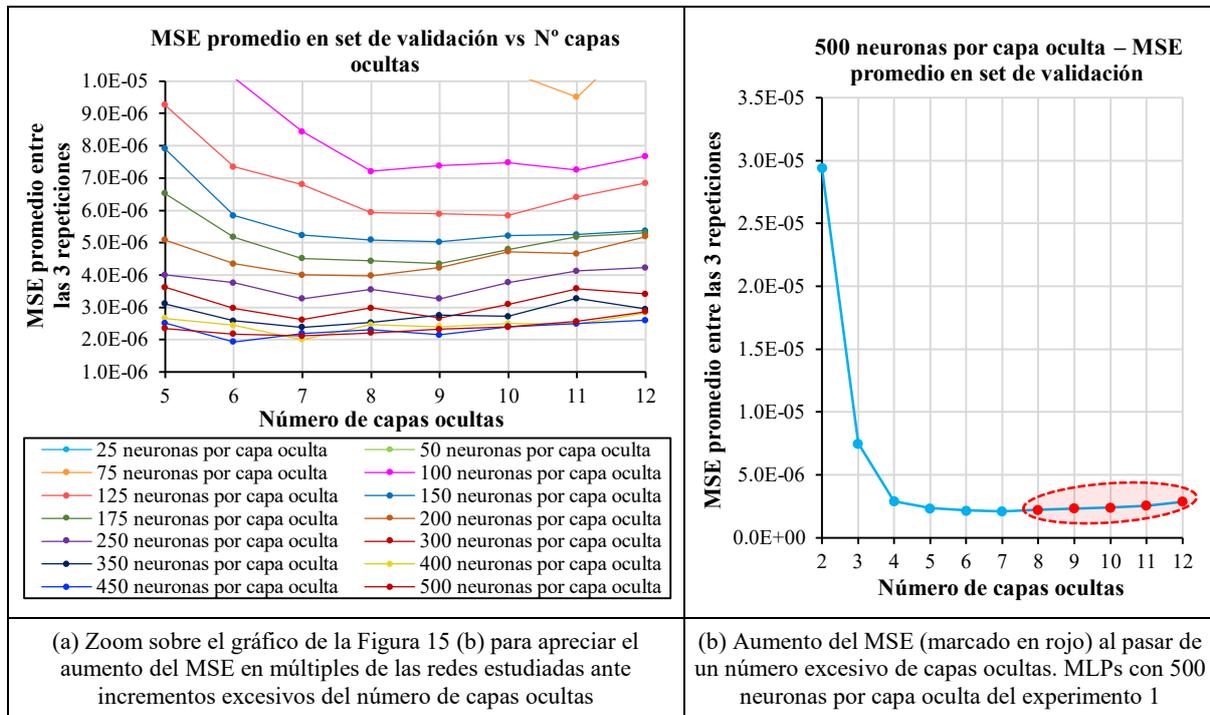


Figura 16: Aumento del MSE promedio en el set de validación al pasar de un número excesivo de capas ocultas.

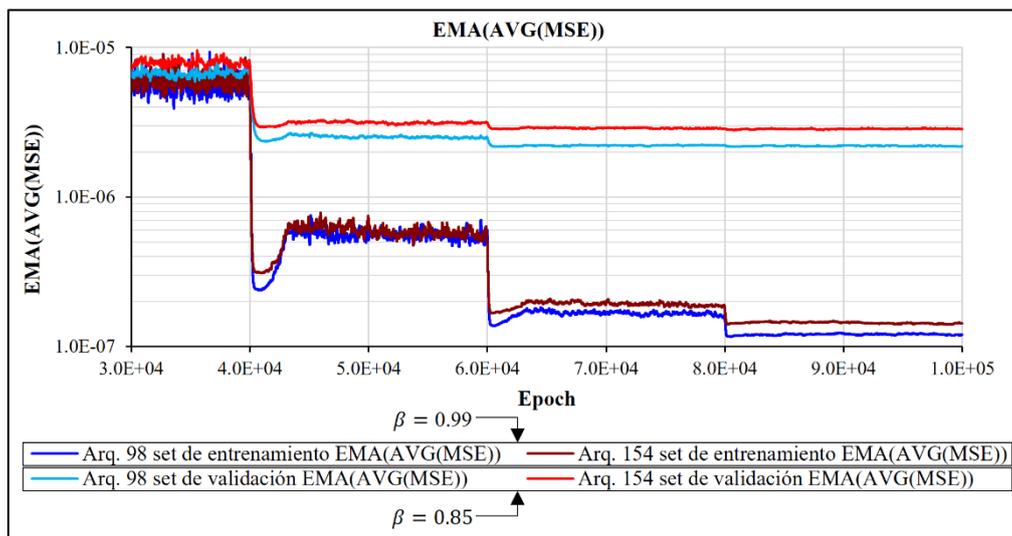


Figura 17: Degradación con el aumento de número de capas ocultas en MLPs con 500 neuronas por capa. Casos representados: 8 capas ocultas (arquitectura 98) y 12 capas ocultas (arquitectura 154).

Es de esperar que la capacidad de generalización disminuya con el aumento del número de capas ocultas en menor medida para los casos con menor número de neuronas por capa oculta. Por lo que se puede afirmar que la tendencia al alza del MSE observada en otros casos estudiados al aumentar excesivamente el número de capas ocultas, se debe también, al menos en parte, a un fenómeno de degradación con el aumento del número de capas ocultas.

Se ha observado que la degradación se da antes (a partir de un número de capas ocultas menor) cuanto mayor es el número de neuronas por capa oculta. De hecho, para las redes estudiadas con 25 neuronas por capa, el número máximo de capas ocultas testeadas no es suficiente como para observar la degradación.

En las superficies del MSE promedio frente al número de neuronas por capa oculta y frente al número de capas ocultas no se aprecian diferencias a nivel macroscópico entre el set de validación y los diferentes sets de test.

Centrando el estudio para los casos con mayor cantidad de parámetros (Figura 18) se aprecia que las superficies obtenidas para los sets de test 3 y 4 son similares en forma a las superficies obtenidas para el set de validación y set de test 1 (ambos tienen sus puntos seleccionados de forma pseudoaleatoria). Por su parte, la superficie obtenida para el set de test 2 (sus puntos fueron distribuidos según una cuadrícula) difiere significativamente de las anteriores. Todo esto permite concluir que, de cara a confeccionar un set con el que seleccionar el número de neuronas ocultas y/o de capas ocultas del MLP, es mejor seleccionar sus puntos de forma pseudoaleatoria

(resultados más parecidos a los de los sets de test 3 y 4). Asimismo, en una etapa del estudio en la que únicamente interese seleccionar los valores para los hiperparámetros, es más óptimo emplear únicamente dos sets: set de entrenamiento y set de validación (con sus puntos seleccionados de forma pseudoaleatoria), prescindiendo del uso de un set de test y seleccionando el valor de los hiperparámetros en base a los resultados para el set de validación.

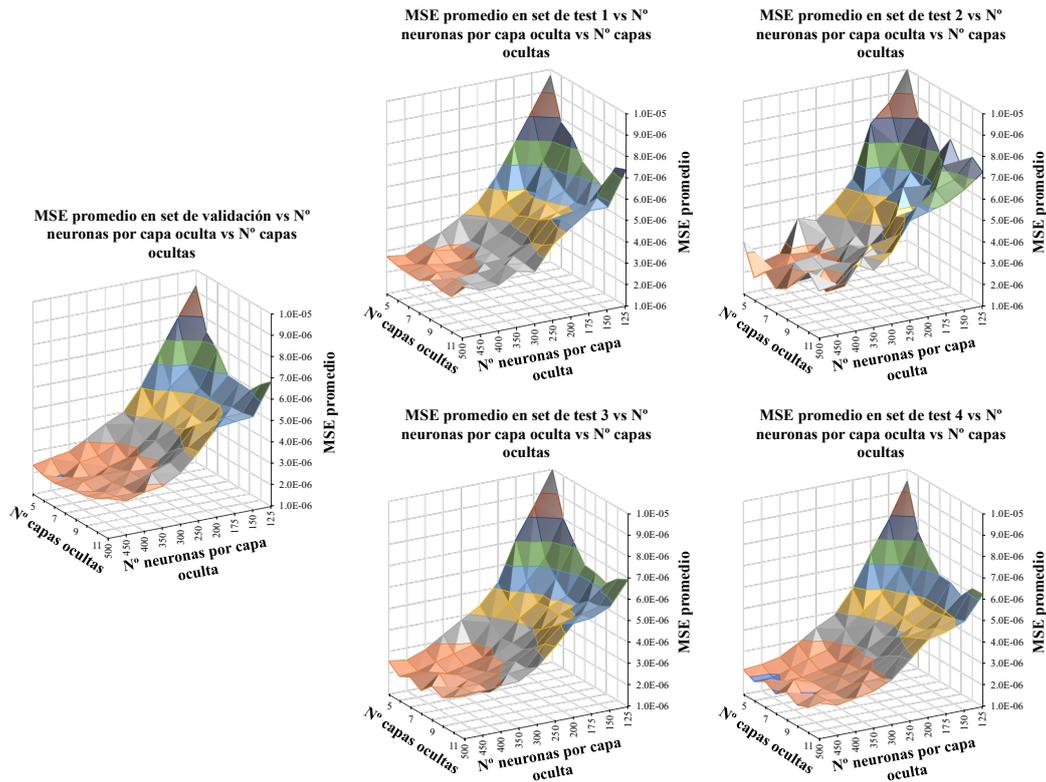


Figura 18: Superficies del MSE promedio frente al número de capas ocultas y frente al número de neuronas por capa. Set de validación y sets de test. MLPs del experimento 1 entre 5 y 12 capas ocultas, y entre 125 y 500 neuronas por capa oculta.

6.3. Selección del número de neuronas por capa oculta y del número de capas ocultas

Un aspecto que puede afectar a la selección del número de neuronas por capa oculta y número de capas ocultas es el tiempo de entrenamiento. En la Figura 19 se puede apreciar, para cada una de las 154 arquitecturas de MLPs del experimento 1, el tiempo de entrenamiento promedio (promediado entre las 3 repeticiones). Se recuerda que todas las redes se han entrenado durante un total de 100,000 epochs para poder asegurar una comparativa de los errores en igualdad de condiciones y para poder analizar posibles casos de overfitting.

Por ello, se señala que existe margen para reducir el número de epochs de los entrenamientos realizados para este artículo, especialmente tras cada una de las disminuciones de la tasa de aprendizaje y, sobre todo, para las redes con menor número de parámetros. En estas últimas, las disminuciones de la tasa de aprendizaje no resultan tan beneficiosas como en redes con un mayor número de parámetros; además, en ellas se alcanza muy rápidamente un plateau de la función de coste tras las disminuciones de la tasa de aprendizaje.

En un caso en el que no interesase estudiar posibles casos de overfitting, resultaría más práctico establecer un criterio de parada diferente, que acompañe o sustituya a un criterio de máximo número de epochs, por ejemplo, un criterio basado en el valor de la función de pérdida en el set de validación.

Con todo lo anterior presente, se concluye que la Figura 19 sirve para comparar el tiempo de entrenamiento a igualdad de epochs, pero que el número de epochs necesarios para cada red depende de multitud de variables, entre ellas, del número de neuronas por capa oculta y número de capas ocultas de la red.

Observando la Figura 19, se aprecian ciertas oscilaciones en los tiempos promedios, debido a efectos de procesos en segundo plano mientras se estaban entrenando las redes. Asimismo, se recuerda que la media se ha extraído de únicamente 3 repeticiones. Debido a estas oscilaciones, en la Figura 19 b) únicamente se han incluido las aproximaciones lineales para los casos de mayor cantidad de número de neuronas por capa oculta (casos de mayores tiempos de entrenamiento, por lo que las fluctuaciones introducidas tienen un menor efecto sobre la tendencia). En el caso de la Figura 19 a) no se han introducido aproximaciones de ningún tipo ya que no se aprecia una tendencia clara (posiblemente por las oscilaciones antes comentadas).

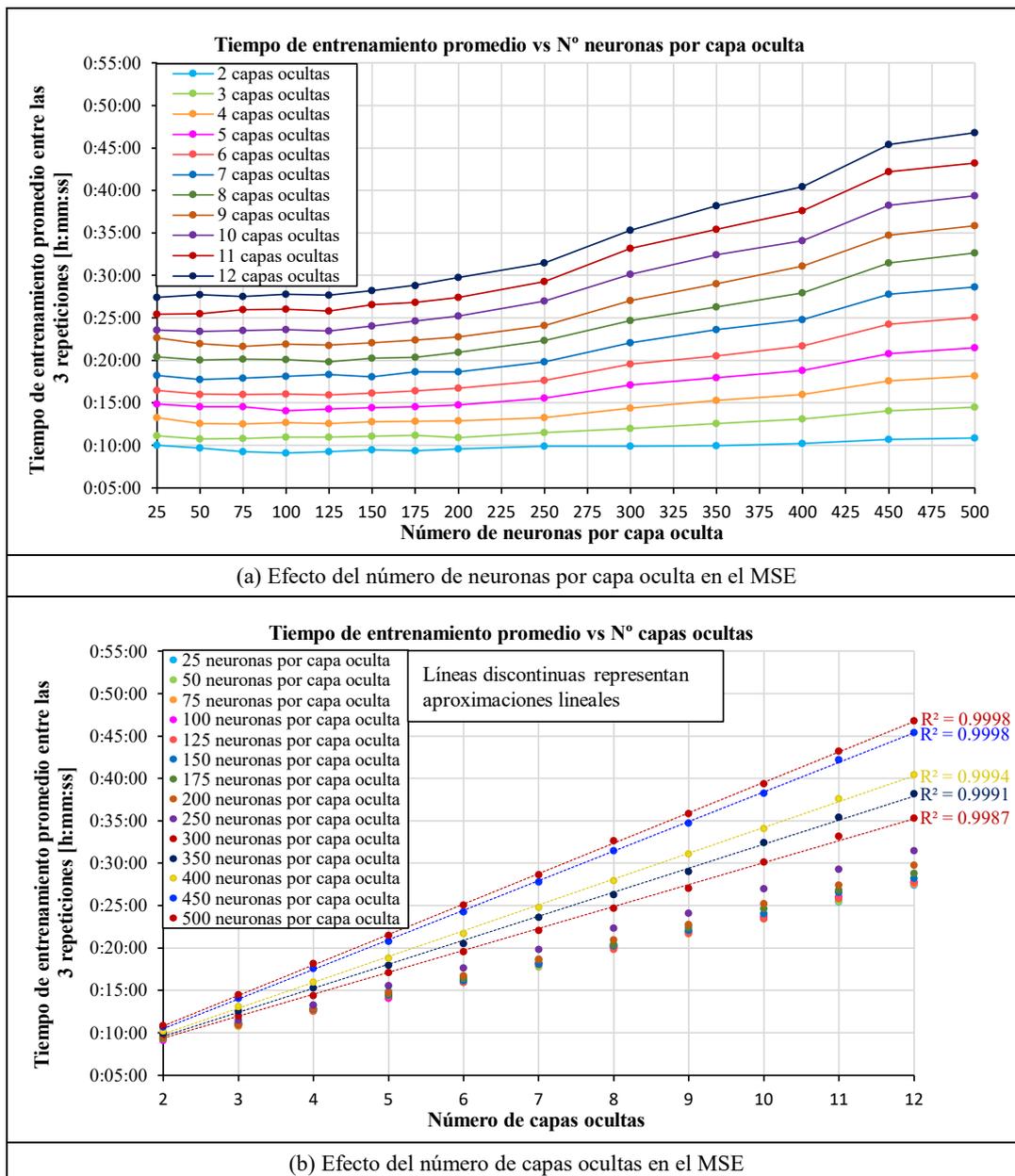


Figura 19: Tiempo de entrenamiento promedio de las 154 arquitecturas de MLPs del experimento 1. Evolución con el número de neuronas por capa oculta y con el número de capas ocultas.

En este artículo no se ha considerado el tiempo de entrenamiento a la hora de seleccionar la arquitectura de MLP más adecuada del experimento 1. De esta forma, dicha selección depende de si se prioriza más la precisión o el coste computacional. La Tabla 2 recoge las arquitecturas que menores MSE promedios y mínimos (entre las 3 repeticiones) proporcionan para los distintos sets de test y para el set de validación:

Tabla 2: Arquitecturas de MLPs del experimento 1 con menores MSE (mínimos y promedios) en los sets de validación y test.

	Hiperparámetros	
	Número de neuronas por capa oculta	Número de capas ocultas
Arquitectura 69	450	6
Arquitectura 70	500	6
Arquitectura 84	500	7
Arquitectura 98	500	8
Arquitectura 111	450	9

Cualquiera de estas redes proporciona un MSE bajo para el problema estudiado (menor a $4.00E-06$, incluso tomando el máximo de los MSE de entre las 3 repeticiones en cualquiera de los sets de test y set de validación). Por tanto, entre las 5 arquitecturas de la Tabla 2 se ha escogido la de menor coste computacional, es decir, la arquitectura 69 (Figura 20).

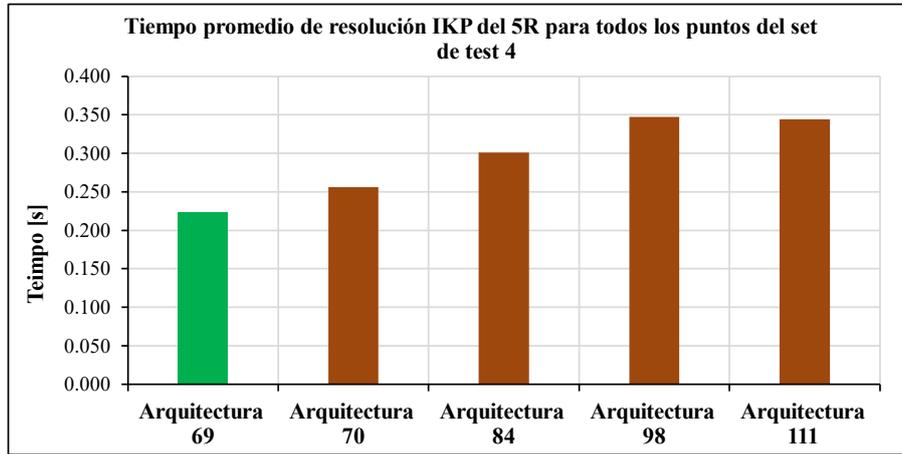


Figura 20: Selección de arquitectura óptima del experimento 1. Tiempo promedio de resolución del IKP del 5R para todos los puntos del set de test 4 usando las arquitecturas de MLPs de la Tabla 2.

7. Resultados del experimento 2

7.1. Efectos en el coste computacional

Para cada uno de los MLPs entrenados en el experimento 2 se ha cronometrado la resolución del 5R para los 15167 puntos del set de test 4, tomando un número suficiente de mediciones como para garantizar tiempos representativos. Al promediar los tiempos de ejecución a lo largo de las 7 repeticiones y para las 3 inicializaciones, se obtienen los resultados de la Tabla 3 (ordenados de menor a mayor coste computacional).

Tabla 3: Tiempo promedio de resolución del IKP completo del 5R para todos los puntos del set de test 4. Resultados del experimento 2.

Coste computacional promedio experimento 2.	
Tiempos promedios, ordenados de menor a mayor	
	(s)
ReLU	0.1759
Sigmoide	0.2247
Swish	0.2329
PReLU	0.2342
tanh	0.2618
Mish	0.3737

Como puede apreciarse en la Tabla 3 y en la Figura 21, la función de activación con menor coste computacional es la ReLU, la siguiente función en cuanto a coste computacional es la sigmoide, la cual presenta un coste computacional ligeramente inferior al de las funciones Swish y PReLU. La función tanh presenta un coste computacional mayor al de la PReLU; mientras que la función Mish tiene un coste computacional significativamente más alto que el del resto de las funciones de activación analizadas. No obstante, se debe tener en cuenta que las diferencias en cuanto a coste computacional entre las funciones de activación pueden verse afectadas en función de cómo estén implementadas éstas.

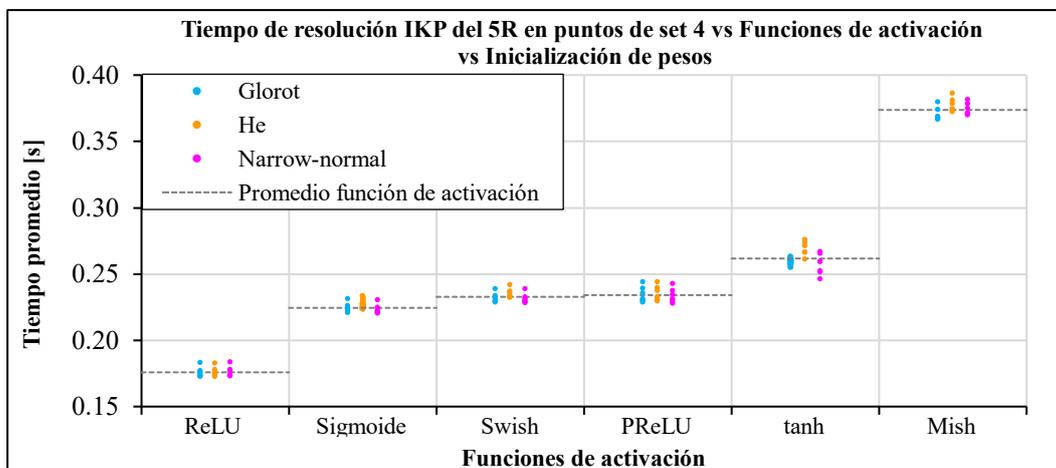


Figura 21: Tiempo de resolución del IKP completo del 5R para todos los puntos del set de test 4. Resultados del experimento 2.

7.2. Efectos en el error de las inferencias

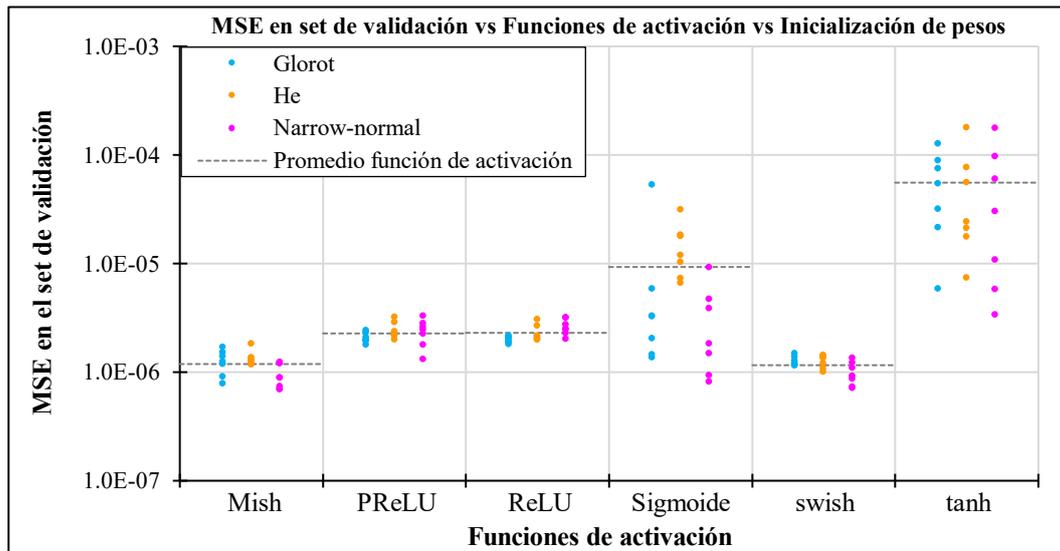


Figura 22: MSE de las inferencias de los MLPs del experimento 2 en el set de validación.

Las funciones de activación Mish y Swish son, en promedio, con las que menor MSE se obtiene de entre las 6 funciones de activación analizadas. Con un MSE promedio ligeramente superior se tiene a la función de activación PReLU, la cual muestra un MSE promedio menor, aunque muy similar, al de la función de activación ReLU. Con un MSE promedio significativamente mayor al de la función de activación ReLU está la función de activación sigmoide. Por último, se tiene la función de activación tanh, que cuyo MSE promedio es significativamente más elevado que el de la función sigmoide.

7.3. Selección de función de activación

La selección de la función de activación más adecuada para la resolución del IKP del 5R depende de si se prioriza la velocidad de procesamiento o la precisión en las inferencias. En el primer caso, la función de activación más adecuada es la función ReLU; mientras que en el segundo caso, las funciones de activación más adecuadas son la función Mish o la función Swish (ambas con MSE promedios similares para el set de validación).

Buscando un equilibrio entre coste computacional y velocidad de procesamiento, las mejores alternativas para este problema son las funciones ReLU y Swish, ya que ambas combinan una velocidad de procesamiento relativamente elevada junto con un adecuado nivel de precisión. En el caso de la función ReLU la balanza se decanta más del lado de la velocidad de procesamiento (siendo la función con mayor velocidad de procesamiento promedio de entre las 6 analizadas y siendo la cuarta función con menor MSE en el set de validación, proporcionando un MSE promedio muy similar al de la función PReLU); mientras que en el caso de la función Swish, la balanza se decanta más del lado de la precisión de las inferencias (siendo la función de activación con un MSE promedio más bajo en el set de validación, aunque muy similar al obtenido con la función Mish, y siendo la función de activación con el tercer menor coste computacional de entre las 6 analizadas). En la Figura 23 se puede apreciar el error absoluto a lo largo del espacio de trabajo para las 4 soluciones del IKP usando, en una función como la representada en la Figura 11, uno de los MLPs con funciones Swish entrenados en el experimento 2.

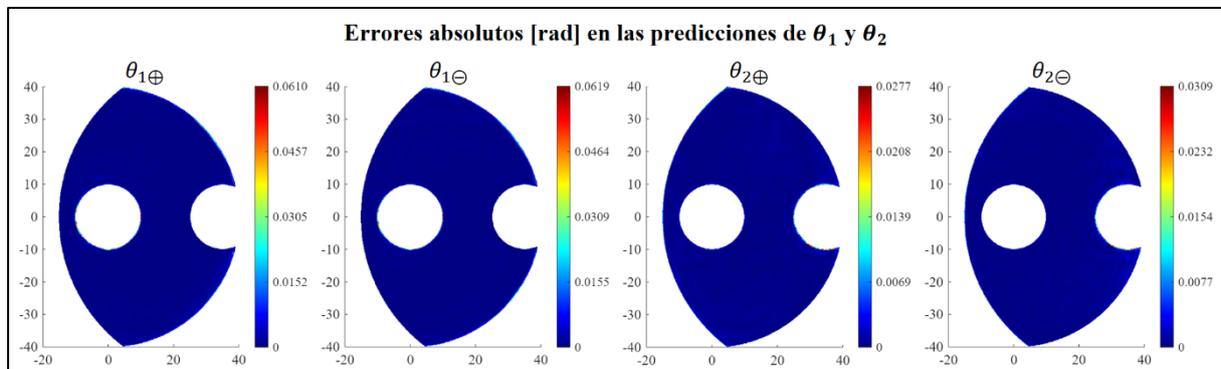


Figura 23: Errores absolutos (en radianes) a lo largo del espacio de trabajo del 5R para las soluciones del IKP obtenidas con uno de los MLPs con funciones de activación Swish entrenados en el experimento 2.

8. Conclusiones

Del experimento 1 se concluye que, al menos dentro de los rangos estudiados de número de capas ocultas y número de neuronas por capa oculta, para MLPs con funciones de activación PReLU, con batch normalization aplicada a las activaciones de cada capa oculta y con un número de neuronas constante a lo largo de sus capas ocultas: el coste computacional crece de forma parabólica con el número de neuronas por capa oculta y de forma lineal con el número de capas ocultas.

Para la mayoría de los MLPs del experimento 1 se observa una degradación de los resultados al sobrepasar un cierto número de capas ocultas. Esta degradación se da antes (a partir de un número de capas ocultas menor) cuanto mayor es el número de neuronas por capa. De hecho, para las redes estudiadas con 25 neuronas por capa, el uso de 12 capas ocultas ha sido insuficiente como para observar la degradación; aunque debe tenerse en cuenta que todos los MLPs analizados cuentan con batch normalization (mitiga la degradación).

Por otra parte, se observa que el MSE de las inferencias tiende (al menos, dentro de los límites de número de neuronas por capa oculta y número de capas ocultas analizados, y con los sets de entrenamiento y validación empleados) hacia asíntotas horizontales conforme aumenta el número de neuronas por capa oculta. Siendo dicha asíntota diferente en función del número de capas ocultas del MLP. No obstante, es esperable que, si se aumentase excesivamente el número de neuronas por capa oculta, apareciesen problemas de overfitting que provocasen un aumento del MSE en el set de validación.

Empleando un criterio que prioriza la precisión de las inferencias, pero que también tiene en cuenta el coste computacional, se ha obtenido que la combinación óptima de número de neuronas por capa oculta y de número de capas ocultas para la resolución del IKP del 5R es de 450 neuronas por capa oculta y 6 capas ocultas.

Se concluye también que, para confeccionar un set con el que seleccionar el número de neuronas ocultas y/o de capas ocultas del MLP, una distribución pseudoaleatoria de los puntos a lo largo del espacio de trabajo es más adecuada que una distribución según una cuadrícula en el plano (x_p, y_p) . Asimismo, en una etapa en la que sólo interesa seleccionar los valores para estos dos hiperparámetros, es más óptimo emplear únicamente dos sets: set de entrenamiento y set de validación (con sus puntos seleccionados de forma pseudoaleatoria), prescindiendo de un set de test y seleccionando el valor de los hiperparámetros en base a los resultados para el set de validación.

Del experimento 2 se concluye que, de menor a mayor coste computacional, las 6 funciones de activación analizadas quedan ordenadas de la siguiente forma: ReLU, sigmoide, Swish configurada de forma que es equivalente a la función SiLU (coste computacional ligeramente superior al de la función sigmoide), PReLU (coste computacional muy similar al de la función Swish), tanh y Mish (coste computacional significativamente mayor al de tanh). No obstante, el coste computacional puede estar condicionado por la implementación de dichas funciones.

De menor a mayor MSE promedio obtenido para el set de validación, estas funciones de activación quedan ordenadas de la siguiente forma: Swish configurada de forma que es equivalente a la SiLU y Mish (ambas con un MSE promedio similar), PReLU (con un MSE promedio ligeramente superior), ReLU (con un MSE promedio muy similar al obtenido con PReLU), sigmoide (con un MSE promedio significativamente mayor que el obtenido con ReLU) y tanh (con un MSE promedio significativamente mayor que el obtenido con sigmoide).

Para la resolución del IKP del 5R, si se busca un equilibrio entre coste computacional y velocidad de procesamiento, las mejores alternativas son las funciones ReLU y Swish. Ambas combinan una velocidad de procesamiento relativamente elevada junto con un adecuado nivel de precisión. Con ReLU se prioriza más la velocidad de procesamiento; mientras que con Swish se prioriza más la precisión de las inferencias.

9. Referencias

- [1] Macho E., Altuzarra O., Pinto C., Hernandez A., “Workspaces associated to assembly modes of the 5R planar parallel manipulator”, *Robotica* **26**, 10.1017/S0263574707004109 (2008).
- [2] Zubizarreta A., Larrea M., Irigoyen E., Cabanes I., Portillo E., “Real time direct kinematic problem computation of the 3PRS robot using neural networks”, *Neurocomputing* **271**, 10.1016/j.neucom.2017.02.098 (2018).
- [3] Zhang Z., Yang L., Sun C., Shang W., Pan J., “CafkNet: GNN-Empowered Forward Kinematic Modeling for Cable-Driven Parallel Robots”, 10.48550/arXiv.2402.18420 (2024)
- [4] Yahia I. B., Merlet J.-P., Papegay Y., “Mixing neural networks and the Newton method for the kinematics of simple cable-driven parallel robots with sagging cables”, *2021 20th International Conference on Advanced Robotics (ICAR)*, 241–246, 10.1109/ICAR53236.2021.9659400, Ljubljana (2021).

- [5] Merlet J.-P., “Advances in the Use of Neural Network for Solving the Direct Kinematics of CDPR with Sagging Cables”, *CabeleCon - 6th International conference on cable-driven parallel robots*, 10.1007/978-3-031-32322-5_3, Nantes (2023).
- [6] Merlet J.-P., “The Inverse Kinematics of Cable-Driven Parallel Robot with More Than 6 Sagging Cables Part 2: Using Neural Networks”, *ARK 2024 - 19th International Symposium on advances in robot kinematics* **31**, 17-24, 10.1007/978-3-031-64057-5_3, Ljubljana (2024).
- [7] Ioffe S., Szegedy C., "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", *Proceedings of the 32nd International Conference on Machine Learning (ICML)* **37**, 10.48550/arXiv.1502.03167, (2015).
- [8] He K., Zhang X., Ren S., Sun J., “Deep Residual Learning for Image Recognition”, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 10.1109/cvpr.2016.90, Las Vegas (2016).
- [9] Kingma D. P., Ba J. L., “Adam: A Method for Stochastic Optimization”, *3rd International Conference for Learning Representations*, 10.48550/arXiv.1412.6980, San Diego (2017).
- [10] Zou H., Hastie T., “Regularization and Variable Selection Via the Elastic Net”, *Journal of the Royal Statistical Society Series B: Statistical Methodology* **67**, 10.1111/j.1467-9868.2005.00503.x (2005).
- [11] He K., Zhang X., Ren S., Sun J., “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 10.48550/arXiv.1502.01852, (2015).
- [12] Ramachandran P., Zoph B., Le Q. V., “Searching for Activation Functions”, 10.48550/arXiv.1710.05941 Preprint at <https://doi.org/10.48550/arXiv.1710.05941>, (2017).
- [13] Elfving S., Uchibe E., Doya K., “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning”, *Neural Networks* **107**, 10.1016/j.neunet.2017.12.012 (2018).
- [14] Misra D., “Mish: A Self Regularized Non-Monotonic Activation Function”, 10.48550/arXiv.1908.08681 Preprint at <https://doi.org/10.48550/arXiv.1908.08681> (2020).
- [15] Glorot X., Bengio Y., “Understanding the difficulty of training deep feedforward neural networks”, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* **9**, Available at: <https://proceedings.mlr.press/v9/glorot10a.html>, Sardinia (2010).