

**UNIVERSIDAD COMPLUTENSE DE  
MADRID**



MÁSTER DE INGENIERÍA DE SISTEMAS Y CONTROL

PROYECTO FINAL DE MÁSTER

BÚSQUEDA DE OBJETIVOS ESTÁTICOS EN TIEMPO  
MÍNIMO MEDIANTE REDES NEURONALES

Alumna: Judith Manso Vergara

Directores: José Antonio López Orozco y Eva Besada Portas

Curso 2012-2013

Convocatoria: Septiembre

# ÍNDICE DE CONTENIDO

---

1	Introducción .....	1
1.1	Motivación .....	1
1.2	Objetivos.....	3
1.3	Aproximación .....	4
1.4	Descripción de la memoria .....	6
2	Antecedentes.....	7
2.1	Aplicaciones reales .....	7
2.2	Estado del arte.....	9
3	Planteamiento del problema .....	12
3.1	El mundo.....	12
3.2	Redes neuronales .....	15
3.2.1	Redes con aprendizaje no supervisado .....	15
3.2.2	Redes de aprendizaje supervisado.....	16
3.3	Heurísticas .....	20
4	Diseño de la red Neuronal.....	23
4.1	Diseño de la red .....	23
4.1.1	Enfoques analizados .....	23
4.1.2	Estudio de los datos conflictivos del entrenamiento .....	32
4.2	Entrenamiento de la red.....	37
4.2.1	Medida de calidad de los resultados .....	37
4.2.2	Ajustes y validación de la red .....	44

4.3	Resultados .....	54
4.3.1	Análisis de los resultados del conjunto de test .....	54
4.3.2	Comparativa de los resultados obtenidos con CEO y con la red neuronal.....	59
5	Conclusiones .....	62
5.1	Consecución de objetivos .....	62
5.2	Trabajo futuro.....	65

## ÍNDICE DE FIGURAS

---

Figura 1: Esquema del proceso de búsqueda de un sistema autónomo ...	2
Figura 2: Camino sub-óptimo con campo de visión reducido.....	4
Figura 3: Esquema de la compensación de par de un cuatrirrotor.....	5
Figura 4: Esquema del mapa de creencia para el caso del submarino USS Scorpion .....	9
Figura 5: Direcciones posibles para el cuatrirrotor .....	12
Figura 6: Representación del campo de visión del agente .....	13
Figura 7: Esquema del controlador a diseñar .....	14
Figura 8: Fases en el entrenamiento de una red no supervisada .....	15
Figura 9: Esquema de funcionamiento de una red supervisada.....	16
Figura 10: Principales funciones de activación en redes neuronales .....	18
Figura 11: Modelo de la red neuronal backpropagation con una capa oculta .....	19
Figura 12: Caminos posibles con sus valores de LET .....	21
Figura 13: Datos correspondientes a una celda en la primera aproximación a la red.....	24
Figura 14: Posibles caminos calculados ignorando las probabilidades más bajas.....	25
Figura 15: Posible camino calculado con un campo de visión reducido ..	25
Figura 16: Gráfica de entrenamiento de la red neuronal primera configuración .....	28
Figura 17: Numeración de las cuadrículas adyacentes al agente .....	29
Figura 18: Esquema de la red neuronal diseñada .....	30
Figura 19: Gráfica de entrenamiento para la aproximación número 2.....	31
Figura 20: Resultados del procesamiento del conjunto de entrenamiento de prueba mediante la red neuronal obtenida en la aproximación 2 .....	32
Figura 21: Caminos posibles en una distribución simétrica.....	34

Figura 22: Distribución de creencias fuera del campo de visión del agente .....	34
Figura 23: Situación en la que menos de una décima parte de las probabilidades están dentro del campo de visión .....	36
Figura 24: Situación en la que menos de una décima parte de las probabilidades están dentro del campo de visión del agente .....	36
Figura 25: Distribución de probabilidad en la que menos de una décima parte de las probabilidades se encuentra dentro del campo de visión del agente .....	37
Figura 26: Comparativa de métricas relacionadas con LET y número de aciertos para un conjunto de datos de test.....	39
Figura 27: Análisis de ejemplo con mejora en el valor de tiempo mínimo estimado.....	40
Figura 28: Análisis de caso en el que la red no recoge suficientes probabilidades .....	41
Figura 29: Casos a ambos lados de la frontera de medidas aceptables..	43
Figura 30: Proceso de ajuste de la red neuronal.....	46
Figura 31: Resultados del procesamiento con la red inicial e los conjuntos de test y entrenamiento .....	48
Figura 32: Evolución de los resultados en el conjunto de validación a lo largo del desarrollo de la red .....	48
Figura 33: Resultados del procesamiento de los conjunto de entrenamiento y validación después de la modificación del parámetro "performance" (actualización 5).....	50
Figura 34: Resultado del procesamiento del conjunto de test con la red neuronal actualizada .....	51
Figura 35: Representación de los resultados obtenidos por la red final para el conjunto de entrenamiento .....	53
Figura 36: Caso de entrenamiento reproducido de forma no eficiente por la red neuronal .....	54

Figura 37: Resultado del procesamiento del conjunto de test "Número de aciertos vs. Tiempo" .....	56
Figura 38: Comparativa de los trazados de red y heurística para el ejemplo con peor resultado: Casotest11 .....	57
Figura 39: Comparativa de las trayectorias calculadas mediante red y heurística en uno de los peores casos: caso de test 48 .....	58
Figura 40: Comparativa de las trayectorias de búsqueda para el cálculo con la red y mediante heurística para el caso de test 2 .....	58
Figura 41: Ejemplo para el que el cálculo efectuado por la red mejora la medida LET: caso de test 47 .....	59
Figura 42: Comparativa del tiempo mínimo calculado mediante red neuronal frente a la heurística .....	60
Figura 43. Tiempos invertidos en el cálculo de trayectorias con la red neuronal.....	61
Figura 44: Comparativa de los cálculos de trayectorias mediante red y heurística con mejoría de la red .....	64

## RESUMEN

---

En este trabajo se aborda el desarrollo de un sistema de control basado en redes neuronales que dirija a un agente, en este caso un cuatrirrotor, para encontrar un objeto estático en tiempo mínimo. La incertidumbre sobre la posición del objeto se modela sobre las celdas de un mapa de probabilidades y la red neuronal se entrena a partir de los datos de entrada (mapa de probabilidades y posición inicial del agente) y salida (secuencia de señales de control) generados por una heurística que minimiza el tiempo esperado de búsqueda. En el trabajo se analiza la viabilidad de la aproximación utilizada sobre múltiples ejemplos, en los que se pone de manifiesto que la red neuronal obtenida es capaz de devolver en tiempo real la secuencia de señales de control que permiten al agente localizar al objeto en un tiempo similar al obtenido con la heurística elegida.

## PALABRAS CLAVE

---

Redes Neuronales, Búsqueda en Tiempo Mínimo, BackPropagation, Entornos con incertidumbre.

## 1 Introducción

En este capítulo se presenta la motivación de este Trabajo Fin de Master, los objetivos perseguidos a lo largo de su desarrollo, y alguna de las ventajas del uso de vehículos aéreos no tripulados en tareas de búsqueda de objetos perdidos.

### 1.1 Motivación

En cualquier tipo de accidente, el tiempo de rescate es un factor decisivo, máxime cuando las condiciones del entorno son críticas para la supervivencia. Este es el caso de los accidentes en el mar donde los naufragos pueden perecer por ahogamiento o congelación.

El riesgo de congelación es un supuesto que cobra especial importancia en los transportes en las zonas frías del planeta ya sean puramente marítimos o aéreos cuando éstos sobrevuelan el mar. Cabe citar el caso de la plataforma Hibernia (Canadá). El 12 de Marzo de 2009, un helicóptero que transportaba trabajadores se precipitó sobre las aguas del Atlántico donde éstas apenas superaban la temperatura de congelación; sólo sobrevivió uno de los dieciocho ocupantes, el resto pasó demasiado tiempo en el agua.

A lo largo de la historia se han sucedido las situaciones en las que se necesita encontrar un objeto perdido con urgencia. En algunas de éstas se han utilizado mapas de probabilidad para orientar las trayectorias de los agentes. En [Stromquist, 2005] se describe el caso de la búsqueda de la bomba nuclear de Palomares en 1966, que tardó 80 días en concluirse. También se emplearon mapas de creencia en el caso del submarino nuclear USS Scorpion en 1968 que se hundió mientras recorría una ruta entre el mediterráneo y Norfolk (Virginia).

Los ejemplos de Palomares y el submarino Scorpion muestran de la importancia que tiene el uso de información probabilista en problemas que involucran incertidumbre.



Para resolver el problema de búsqueda, es necesario calcular el plan óptimo de guiado, que es el orden en el que el agente debe visitar las diferentes áreas geográficas para encontrar el objetivo en el tiempo mínimo posible.

En la actualidad las tareas de búsqueda son llevadas a cabo por humanos, pero gracias a los avances tecnológicos que ha permitido miniaturizar sensores y actuadores así como avanzar en la capacidad de cálculo de los sistemas empotrados, estas tareas podrán ser llevadas a cabo por sistemas autónomos en un futuro.

La automatización del proceso de búsqueda implica la creación de un sistema de recogida y procesamiento de la información del entorno, y de un controlador que calcule las trayectorias. Este trabajo se va a centrar en la parte del controlador que calcula las trayectorias.

Un esquema básico del proceso se muestra en la Figura 1:



Figura 1: Esquema del proceso de búsqueda de un sistema autónomo

Históricamente son muchos los investigadores que han encarado el problema de la detección de objetivos perdidos, en la mayoría de los casos olvidándose del objetivo fundamental: minimizar el tiempo de búsqueda [Stone, 1975]. Lo habitual es seguir la segunda aproximación mencionada en [Trummel et al, 1986]: maximizar la probabilidad de detección. A pesar de que ambas opciones son

factibles, no alcanzan los mismos resultados ya que en entornos con restricciones en la dinámica del agente no son equivalentes.

En [Lanillos, 2013] se presenta un método de búsqueda de caminos sub-óptimos mediante algoritmos de optimización probabilísticos. El estudio del presente proyecto trata del diseño de un controlador que calcule trayectorias sub-óptimas con los bajos requerimientos de memoria, alta adaptabilidad, propiedades de extrapolación y capacidad de respuesta en tiempo real que nos aportan las redes neuronales en base a los resultados obtenidos por el algoritmo propuesto por Pablo Lanillos [Lanillos, 2013].

## 1.2 Objetivos

El objetivo de este trabajo consiste en el diseño de un controlador en tiempo real que calcule las señales de control sub-óptimas que hay que aplicar al agente para que siga la trayectoria de búsqueda. Los datos de entrada al controlador serán mapas de creencia. Este controlador está pensado para integrarse en un cuatrirrotor para su guiado en tareas de búsqueda.

El controlador calculará una secuencia de  $N$  señales de control, asumiendo un objeto estático cuya posición no varía con el tiempo.

Las trayectorias a calcular deben minimizar el tiempo para localizar el objetivo (Minimum Time Search, MTS) ya que la aproximación de maximizar la probabilidad de detección no implica la minimización del tiempo en caminos con restricciones dinámicas.

El horizonte de visión del agente tiene que ser suficientemente amplio como para permitirle tomar un buen camino en los  $N$  movimientos que calcule. Esto es, por una parte, si proporcionamos un número insuficiente de datos, el agente va a tomar las decisiones en base a una información incompleta. Por otro lado, si proporcionamos más información de la necesaria estamos complicando innecesariamente el problema del cálculo de trayectorias. Esto se puede ver en el

ejemplo de la Figura 2, en el que las probabilidades están coloreadas según importancia en orden ascendente [verde, amarillo, rojo]. Para el horizonte de visión, el mejor camino es el señalado con las flechas, sin embargo, si nuestra visión no estuviese limitada, el camino podría recoger las probabilidades marcadas en rojo.

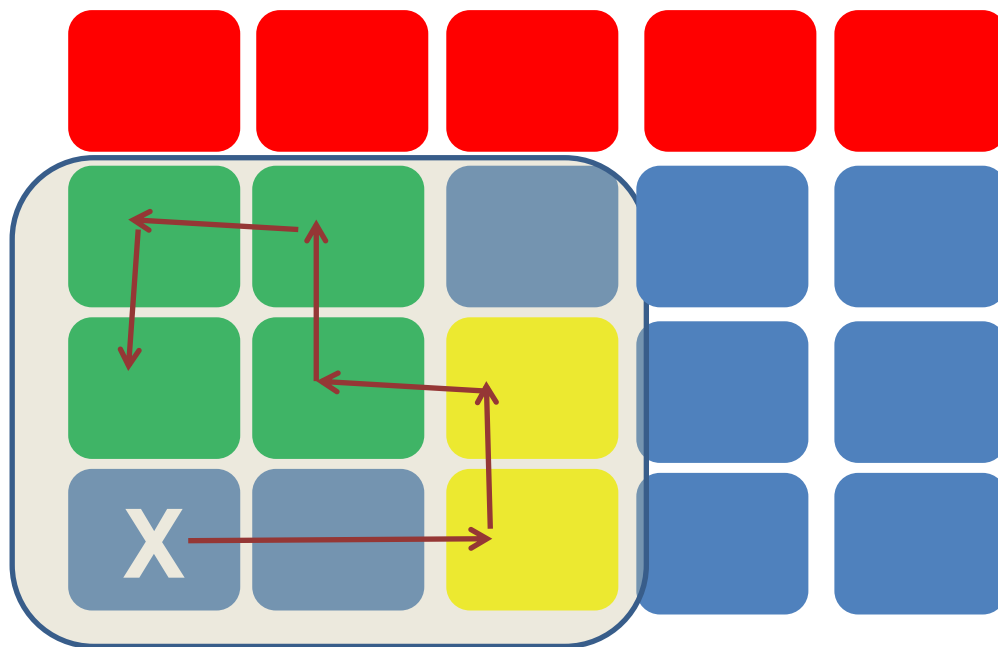


Figura 2: Camino sub-óptimo con campo de visión reducido

Partiendo del conocimiento adquirido por el algoritmo propuesto en [Lanillos, 2013] se busca diseñar una red neuronal que aporte al controlador capacidad de respuesta en tiempo real, bajo los requerimientos de memoria y las propiedades de extrapolación que se necesitan para poder integrar el sistema en un vehículo aéreo no tripulado (Unmanned Aerial Vehicle, UAV).

### 1.3 Aproximación

El UAV se define como un vehículo propulsado por un motor de reacción, sin tripulación, reutilizable y capaz de mantener un vuelo controlado. El factor que diferencia los UAV de los misiles es la capacidad de reutilización de los primeros [Berrú Correa, 2013].

El controlador que se diseña en este estudio debe estar orientado para su uso en un cuatrirrotor. Un cuatrirrotor es un tipo de UAV con características más adecuadas que las de un helicóptero convencional para tareas de búsqueda no tripulada.

En un helicóptero convencional la rotación es controlada por el rotor de cola, en el caso de los cuatrirrotores, el giro de dos de sus rotores en sentido horario y de los otros dos en antihorario compensará la rotación (ver Figura 3). De esta forma el vehículo se mantiene estable. Además, la carencia de rotor de cola elimina la elevación innecesaria que este provoca.

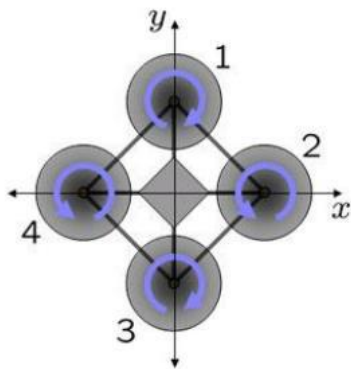


Figura 3: Esquema de la compensación de par de un cuatrirrotor

El empleo de cuatrirrotores como UAVs tiene varias ventajas frente al uso de helicópteros convencionales a escala, la principal es la reducción en el diámetro de los rotores (consecuencia directa de la sustitución de un rotor en los helicópteros convencionales por cuatro en el caso que nos ocupa). Ésto disminuye el daño que puede causar el choque accidental de los rotores con cualquier obstáculo, haciendo a los cuatrirrotores unos vehículos seguros para interactuar en áreas reducidas.

## 1.4 Descripción de la memoria

Este trabajo está estructurado en torno a cuatro bloques principales, (antecedentes, aproximación al problema, desarrollo y conclusiones), cada uno recogido en uno de los siguientes capítulos de la memoria.

En antecedentes (capítulo 2) se recoge una breve introducción acerca de los trabajos y aplicaciones reales que se han enfocado a la búsqueda de objetos a lo largo de la historia y que nos han llevado a realizar este estudio.

En el planteamiento del problema (capítulo 3) se explican los datos de partida para desarrollar el controlador basado en redes neuronales. Este punto también incluye una introducción de los conceptos sobre los que se sustenta este trabajo: redes neuronales y el algoritmo de optimización probabilístico de entropía cruzada.

En el bloque de diseño de la red neuronal (capítulo 4) se describe todo el proceso de decisión llevado a cabo para elegir la configuración de la red. También se fijan los objetivos que tiene que cumplir y se hace un seguimiento de la evolución de éstos en las diferentes etapas de refinado. Al finalizar este apartado se realiza una comparativa de los resultados obtenidos con la red neuronal frente a los que nos proporciona el algoritmo de optimización probabilístico de entropía cruzada.

El último bloque (capítulo 5) está destinado a las conclusiones, en él se repasa la configuración obtenida para la red y el cumplimiento de los objetivos. Este bloque termina con una sección dedicada a trabajos futuros para seguir avanzando en la línea de este estudio.

## 2 Antecedentes

Antes de abordar el desarrollo del controlador, es necesario dar un breve repaso a los antecedentes que han motivado este estudio.

### 2.1 Aplicaciones reales

La primera aplicación real de búsqueda basada en mapas de probabilidad tiene lugar en el marco de la guerra fría, en el año 1966 en Palomares, en el que un bombardero estratégico procedente de Carolina del Norte que transportaba cuatro bombas de hidrógeno colisionó con un avión cisterna durante el repostaje. Horas después del accidente tres de las cuatro bombas se recuperaron, pero la cuarta seguía desaparecida a pesar de las búsquedas intensivas en el área.

En los primeros días de búsqueda, una división de las fuerzas armadas de los Estados Unidos tomó fotografías aéreas de la zona del accidente para tener el mayor número de datos posible. Después de esta etapa, las fuerzas armadas pidieron ayuda a las fuerzas Navales de EEUU, quienes contrataron a un comité de expertos para solucionar el problema.

La búsqueda se llevó a cabo mediante un novedoso método matemático: la teoría de búsqueda de Bayes. Este método asigna probabilidades a las casillas individuales de un grid que representa el espacio de búsqueda de forma discretizada y va actualizando los valores a medida que la búsqueda progresa. Como este método requiere una asignación inicial de probabilidades, se pidió ayuda a un pescador local que pudo ver cómo la bomba entraba en el agua. La cuarta bomba se encontró pasados 80 días de accidente.

La guerra fría fue también desencadenante de otro accidente que implicó la utilización de mapas de probabilidad para búsqueda de objetos en entornos con incertidumbre [Wagner, 2010]. En este caso se trata del hundimiento del

submarino nuclear USS Scorpion en Mayo de 1968 mientras recorría una ruta desde el mediterráneo hasta Norfolk (Virginia).

El método de búsqueda utilizado fue el siguiente:

1. Toma de datos con sónar. Los datos adquiridos en este punto suponen el punto de partida para los mapas de probabilidad. En la Figura 4 se observan las señales del sónar, así como la localización real del Scorpion y la casilla cuya probabilidad era más elevada.
2. Construcción de mapas de probabilidad mediante simulación Montecarlo. Se tiene incertidumbre en la localización del accidente y en las características del movimiento. Se definen nueve escenarios para los que se realiza una simulación de Montecarlo.
3. Cálculo de la probabilidad de efectividad local. Dado que los detectores no son ideales y que las celdas tienen unas dimensiones de 1x0,84 millas, el hecho de pasar por la celda donde se encuentra el submarino no implica encontrarlo. En este caso la probabilidad de detección del objeto dentro de una celda es del 4%.
4. Estimación de la probabilidad de éxito de un plan de búsqueda propuesto.
5. Actualización del mapa de probabilidad con las celdas ya exploradas.

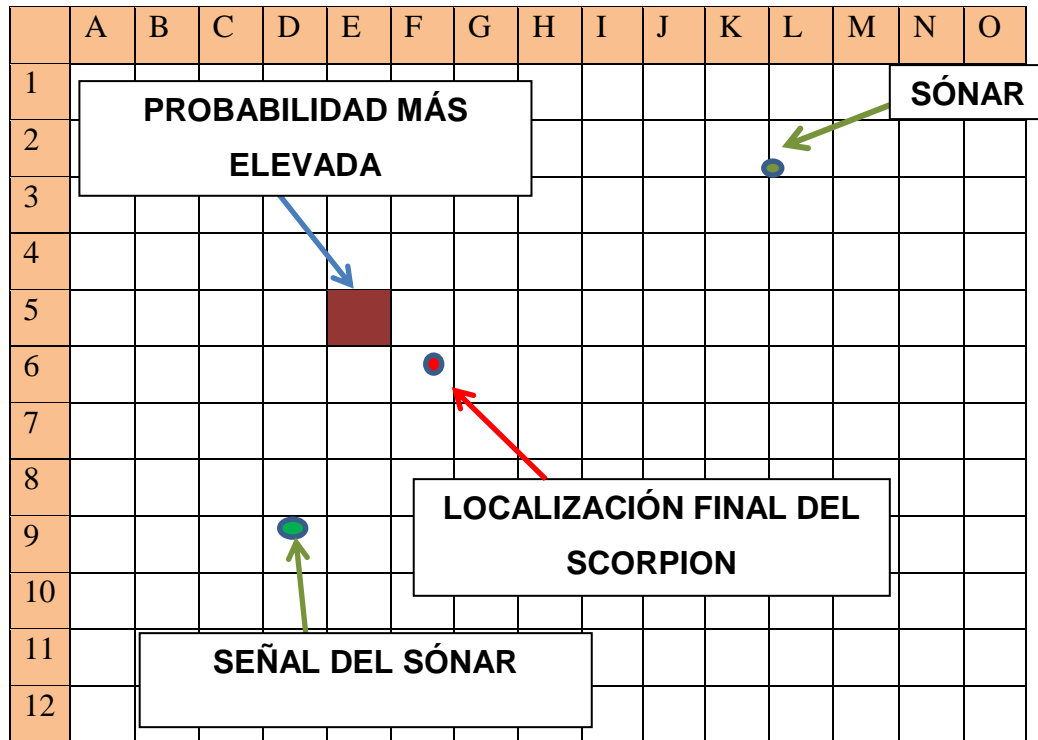


Figura 4: Esquema del mapa de creencia para el caso del submarino USS Scorpion

Finalmente el submarino se encontró 5 días después de comenzada la búsqueda, en una localización que estaba a tan sólo 200 metros del área de probabilidad más elevada.

La teoría probabilística fruto del trabajo de Bayes y sus contemporáneos, la teoría de algoritmos y una disciplina que nació durante la guerra fría basada en la mejora en la toma de decisiones en base a métodos matemáticos dieron forma a lo que años más tarde se denominaría “La búsqueda óptima”. En el libro “Teoría de la Búsqueda óptima” [Stone, 1975] se puede encontrar un compendio de los métodos empleados por la marina en la búsqueda de objetos perdidos.

## 2.2 Estado del arte

Los trabajos relacionados con la búsqueda de objetos en entornos con incertidumbre se pueden clasificar en tres períodos de investigación.

El primero (años 1970-1984) los trabajos estaban condicionados a las limitaciones de cálculo de los ordenadores de la época. En esta época cabe



reseñar el trabajo de Stone [Stone, 1975] en el que centra el problema en calcular la distribución óptima de tiempo dedicada a explorar cada región de búsqueda. Durante este período, algunos autores se dieron cuenta de que la búsqueda era un caso particular de los procesos de decisión parcialmente observables de Markov (Partially Observable Markov Decision Process, POMDP) ya que al final se utilizan observaciones parciales de un estado desconocido: la localización del objetivo. Siguiendo esta línea, en 1984, Eagle [Eagle, 1984] demostró que si se introducían restricciones en el trazado de la búsqueda los métodos de Stone no funcionaban. Entonces, desarrolló un método para resolver el problema de la búsqueda cuando el camino está restringido, basándose en algoritmos de programación dinámica para resolver POMDPs. Como la complejidad del problema depende exponencialmente del número de celdas del mapa y la capacidad de computación de los ordenadores disponibles era baja, tan solo se resolvieron problemas en escenarios pequeños.

En un segundo período Eagle, Yee y Washburn [Eagle et al, 1990; Washburn, 1998] diseñaron algoritmos que redujeron el tiempo de computación limitando el número de estados viables, utilizando una nueva idea, clave para futuras líneas de investigación: si el problema consiste en encontrar un objetivo perdido, éste termina cuando el objetivo es encontrado, convirtiendo al problema POMDP en una optimización determinista donde la observación realizada es únicamente “no detección”.

El último período comienza en los albores del siglo XXI, cuando un grupo de investigadores que trabajaban en fusión sensorial reactivaron el desarrollo de la búsqueda óptima. Los algoritmos existentes desarrollados en este período se pueden organizar en los siguientes grupos:

- Nuevos algoritmos que siguen la corriente iniciada con los POMDPs. [Kaelbling et al., 1998; Hsu et al., 2008]

- Nuevos algoritmos heurísticos que formulan el problema de búsqueda de forma discreta optimizando la entropía [Yang et al., 2002, 2004] y el tiempo de búsqueda [Sarmiento et al, 2009; Lanillos, 2013].
- Aproximación al problema desde una perspectiva continua optimizando la probabilidad de detección para un paso con un algoritmo voraz [Bourgaut et al, 2003, 2004] y para N pasos con un algoritmo basado en gradiente [Mathews et al, 2007; Gan, 2010; Lanillos, 2013].
- Técnicas alternativas en las que se utilizan otras representaciones probabilísticas del estado de los objetos y la búsqueda se realiza basándose en otras técnicas. Dentro de este grupo queremos destacar los Trabajos de Fin de Máster [Fernández Ramírez, 2010; Sáez Martínez, 2012] en los que se usan redes neuronales para predecir el estado de los objetos y para obtener las señales de control del agente, y la publicación [Fernández Ramírez et al, 2011] en la que se sustituye la red neuronal de predicción por un filtro de partículas.

En este trabajo se estudia la viabilidad del uso de redes neuronales como generadores de trayectorias de tiempo mínimo dado un mapa probabilístico con el objeto de combinar la eficiencia computacional de las redes neuronales y el objetivo de tiempo mínimo de los algoritmos discretos recogidos en [Lanillos, 2013].

### 3 Planteamiento del problema

En este capítulo se describen la forma en que entendemos el mundo y las bases sobre las que se apoya este estudio: redes neuronales y algoritmo heurístico CEO.

#### 3.1 El mundo

El mundo está formado por un mapa que contiene las probabilidades de encontrar un objeto en cada celda del mapa, de forma que la probabilidad total de todas las casillas suma uno.

El agente que va a realizar la búsqueda se trata de un cuatrirrotor. Como ya se expuso en los antecedentes, este dispositivo es un UAV.

El cuatrirrotor recibirá las señales calculadas por el controlador objeto de este estudio. Estas señales se traducen en un array de movimientos individuales de una unidad o cuadrícula codificados según el ángulo de desplazamiento como se muestra en la Figura 5.

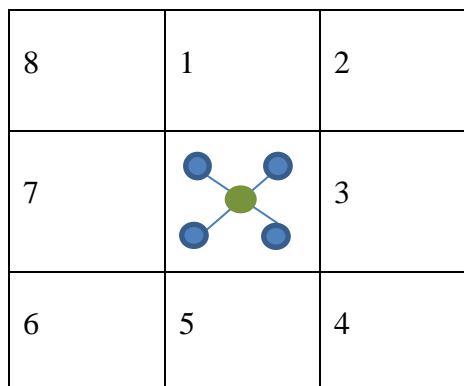


Figura 5: Direcciones posibles para el cuatrirrotor

El número de secuencias de control que se desea calcular es  $N=10$  ya que este valor supone un avance significativo en la trayectoria del cuatrirrotor, tras el cual el agente puede volver a tomar una imagen de las probabilidades de su entorno y calcular una nueva secuencia.

El horizonte de visión comprende un cuadrado de lado 21 unidades en cuyo centro se encuentra el agente. El mapa de la Figura 6 representa esquemáticamente en amarillo las celdas en el horizonte de visión y en gris las celdas que no influyen en los cálculos de la trayectoria.

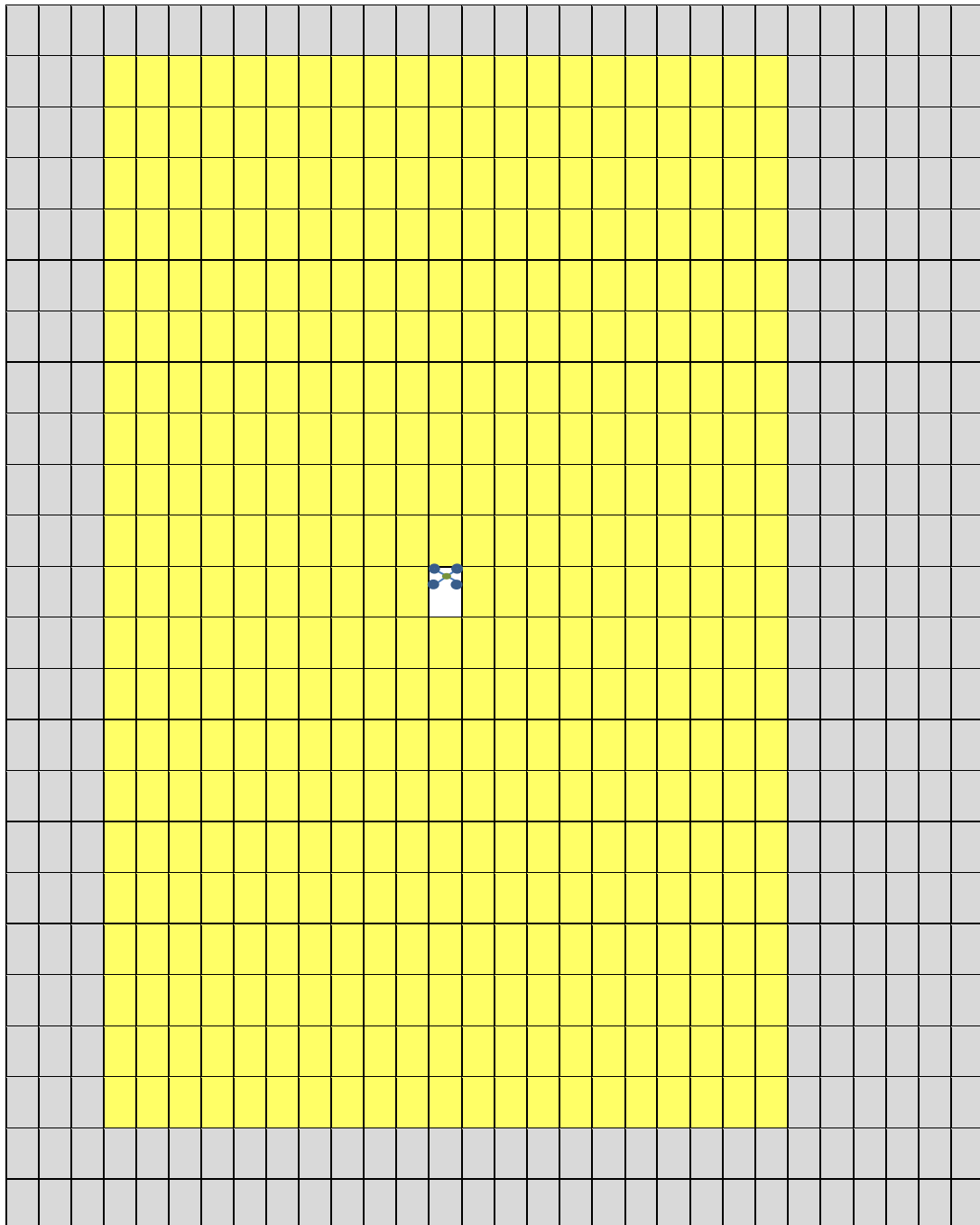


Figura 6: Representación del campo de visión del agente

En este proyecto se aborda el problema de búsqueda de objetos estáticos, en el que el mapa de probabilidades no se ve modificado por el movimiento del agente aunque sí por las observaciones realizadas por él. Además, en este estudio se considera que el agente no detecta al objeto hasta que finaliza la búsqueda y se modela el comportamiento del sensor de forma ideal, de manera que la probabilidad de una celda observada por el agente en la que no se detecta el objeto pasa a tener valores nulos.

El esquema del controlador que en un futuro se quiere integrar en un cuatrirrotor para el cálculo de las trayectorias de búsqueda en tiempo real es el que se muestra en la Figura 7. En ella, el controlador está representado por una red neuronal, ya que el objetivo principal de este proyecto es determinar la viabilidad del uso de redes neuronales para obtener la trayectoria de búsqueda de un agente sobre un mapa de probabilidad.

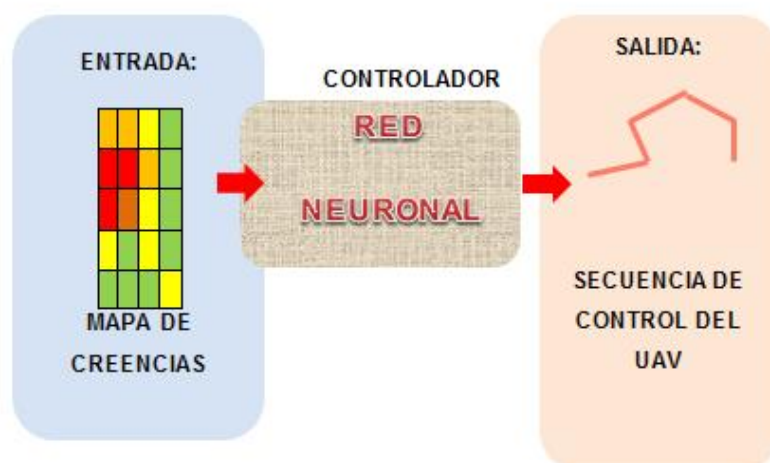


Figura 7: Esquema del controlador a diseñar

## 3.2 Redes neuronales

Las redes neuronales pueden clasificarse en dos grandes grupos atendiendo al modelo de aprendizaje que emplean: supervisado y no supervisado.

### 3.2.1 Redes con aprendizaje no supervisado

Las redes no supervisadas buscan el grado de similitud entre los diferentes datos que procesan. El aprendizaje de estas redes sólo requiere un conjunto de entradas. En el proceso de entrenamiento de un modelo no supervisado: se alimenta la red con una serie de entradas (en el ejemplo de la Figura 8 colores) y ésta calcula la diferencia entre los valores de las propiedades de entrada y extrae los tipos de elementos existentes (tipo1, tipo2 y tipo3 en el ejemplo).



Figura 8: Fases en el entrenamiento de una red no supervisada

Una de las tipologías más populares de aprendizaje no supervisado son los mapas auto organizativos (Self-Organizing-Map, SOM), donde a partir de unos datos de entrada, la red obtiene un mapa en el que los elementos atienden a una clasificación o clustering definido en el entrenamiento.

Las aplicaciones principales de las redes de aprendizaje no supervisado son la clasificación y compresión de datos. En el problema que nos ocupa los datos de partida están distribuidos en un grid, sin embargo, el propósito de la red que necesitamos no es clasificar sino predecir una ruta sub-óptima.

### 3.2.2 Redes de aprendizaje supervisado

Las redes de aprendizaje supervisado se entrenan mediante pares entrada-salida con un objetivo: ser capaces de procesar una entrada para obtener la salida dada. Esto es posible mediante el ajuste de los pesos en las capas internas de la red [Pajares et al, 2007].

En la Figura 9 se explica esquemáticamente el funcionamiento del aprendizaje supervisado. La red se alimenta con pares entrada-salida y a lo largo del entrenamiento simula la respuesta ante las diferentes entradas. A la salida existe un maestro que evalúa si la respuesta es correcta o no y en el caso de no serlo, los pesos de las capas internas de la red neuronal se reajustan.

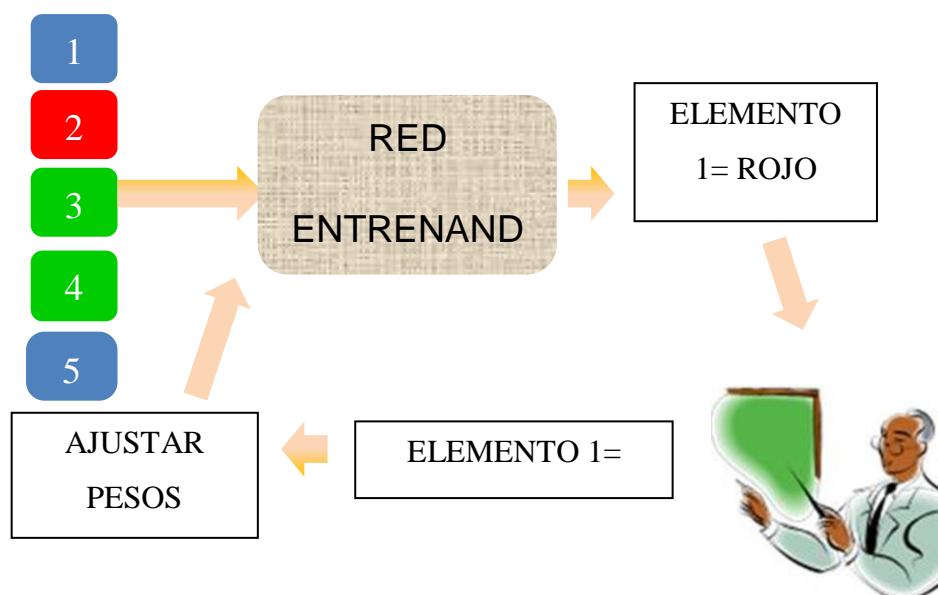


Figura 9: Esquema de funcionamiento de una red supervisada

Las redes más populares con aprendizaje supervisado son retropropagación (backpropagation) y base radial.

La red backpropagation tiene una arquitectura en niveles y las conexiones entre neuronas van hacia delante de modo que cada neurona de una capa proporciona la entrada a otra neurona en la capa siguiente. El entrenamiento de estas redes comienza con la asignación aleatoria de un patrón de pesos que propaga el valor de las entradas a lo largo de todas las capas de la red hasta llegar a la salida donde los valores de los nodos de salida son comparados con las respuestas deseadas para generar los términos de error. Los errores obtenidos a la salida se transmiten hacia atrás para adaptar los pesos de los diferentes nodos.

La salida de una neurona se evalúa en función del valor de activación de la misma. Este valor se obtiene procesando el valor de la entrada mediante la función de transferencia de la neurona. Así, en una neurona con función de transferencia escalón acotada  $[0, 1]$  los valores inferiores a cero darán una respuesta 0 (neurona inactiva), y los valores positivos activarán la neurona. El comportamiento de las funciones de transferencia habituales (escalón, lineal, sigmoide y gaussiana) se presenta en la Figura 10.



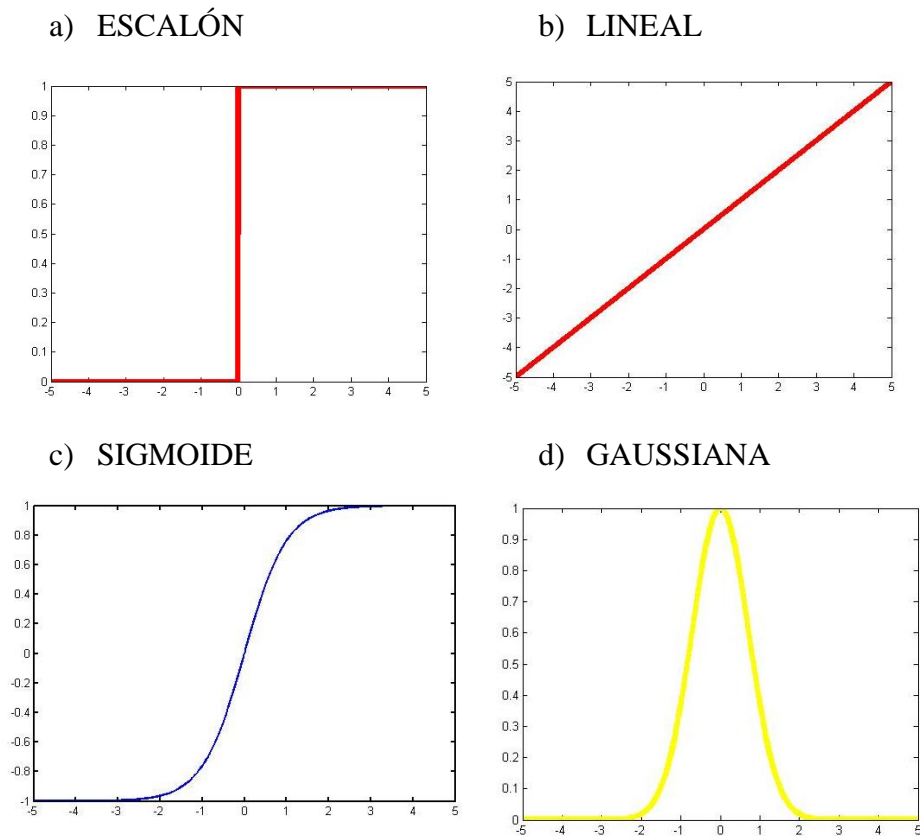


Figura 10: Principales funciones de activación en redes neuronales

La red backpropagation tiene un mínimo de dos capas de neuronas: una capa de entrada cuyo tamaño coincide con el número de entradas a la red y una capa de salidas que coincide con el número de salidas deseadas. En las redes de mayor complejidad es posible introducir capas ocultas intermedias para facilitar la convergencia. En la Figura 11 se muestra esquemáticamente la arquitectura de una red de 3 capas de neuronas.

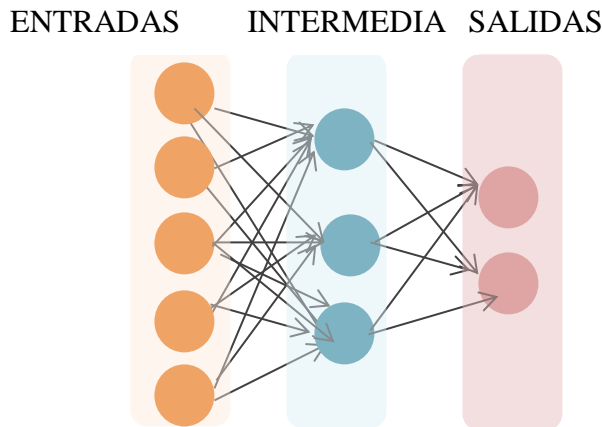


Figura 11: Modelo de la red neuronal backpropagation con una capa oculta

Los aspectos fundamentales que diferencian la red de base radial de backpropagation son la limitación en el número de capas (en la red de base radial solo podemos disponer de 3 capas de neuronas) y la elección de función de transferencia de la capa intermedia (en la red de base radial esta función de transferencia es gaussiana).

Con todo lo expuesto hasta ahora, se hace evidente que la red que necesita el problema que nos ocupa es de tipo supervisado pudiendo ser igualmente válida una red de retropropagación o una de base radial. Con el objeto de dar continuidad al trabajo realizado anteriormente [Sáez Martínez, 2012] en el que una red neuronal backpropagation se utilizaba para coordinar las labores de rescate de un conjunto de UAVs, en este proyecto se utilizarán redes del mismo tipo.

Como la red que se empleará es supervisada, es necesario generar los pares entrada-salida de entrenamiento. Para generar estos pares de datos en los que las entradas son mapas de probabilidades y las salidas son secuencias de N señales de control se utilizará el algoritmo heurístico propuesto en [Lanillos, 2013], que se explica con detalle en el siguiente punto.

### 3.3 Heurísticas

El generador de pares entrada-salida es una heurística implementada por Pablo Lanillos Pradas en su tesis: “*Búsqueda de objetivos móviles en tiempo mínimo sobre entornos con incertidumbre*” [Lanillos, 2013]. En este punto se explicará brevemente en qué consiste el algoritmo.

El cálculo de las trayectorias subóptimas de la heurística parte de una aproximación discreta al problema de búsqueda en tiempo mínimo donde el espacio está representado por una cuadrícula cuyas celdas tienen asignada una probabilidad de encontrar el objetivo.

El objetivo del algoritmo es minimizar el tiempo local esperado de búsqueda (Local Expected Time, LET). El cálculo de dicho tiempo se determina en la tesis truncando la serie calcula probabilísticamente el tiempo esperado de detección.

La forma de calcular dicha función objetivo para el caso estático con un sensor ideal se ilustra en el ejemplo recogido en la Figura 12. La idea subyacente es que este tiempo se puede calcular como la probabilidad de no encontrar el objeto hasta el primer paso, más la probabilidad de no encontrarlo hasta el paso dos y así sucesivamente, teniendo en cuenta que las probabilidades de las celdas visitadas son nulas. Dicho de otra forma, el valor de LET se calcula como la probabilidad restante acumulada a lo largo de todas las trayectorias. Como se puede ver en la Figura 12 para una misma distribución de probabilidad, la elección del camino varía significativamente el valor de esta medida.

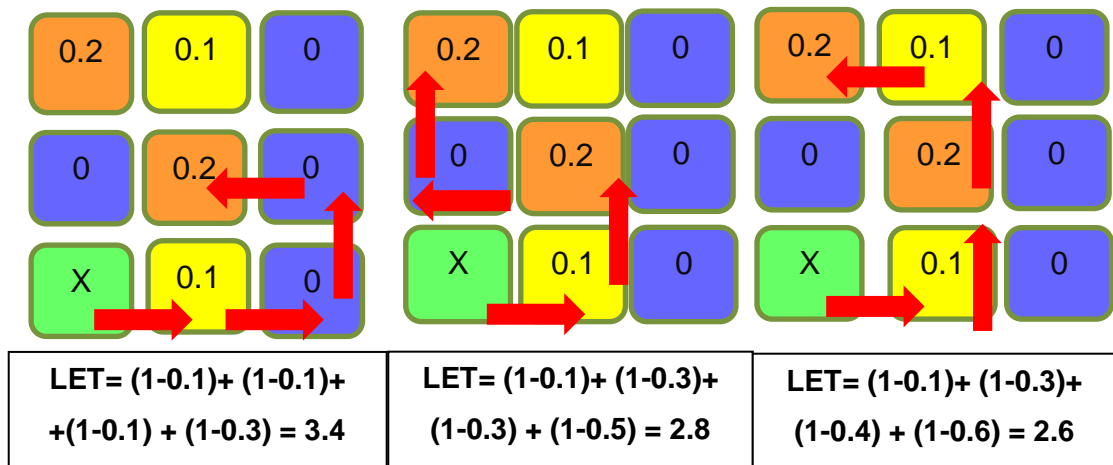


Figura 12: Caminos posibles con sus valores de LET

El método utilizado para optimizar la función LET es el algoritmo iterativo de optimización probabilístico de entropía cruzada (Cross Entropy, CEO).

Al igual que en el controlador de la red neuronal que queremos desarrollar, la entrada al algoritmo es el mapa de probabilidades y su salida la mejor secuencia de acciones para un número N finito de iteraciones.

El algoritmo utiliza una población de posibles soluciones (secuencia de acciones) sobre las que se realizan las siguientes operaciones:

- 1) Evaluar la función objetivo (LET).
- 2) Determinar el subconjunto de mejores M soluciones.
- 3) Calcular la probabilidad de realizar la acción en cada uno de los N pasos de la secuencia de acciones, a partir de los datos del subconjunto anterior.
- 4) Utilizar las probabilidades obtenidas en el paso 3 para muestrear el nuevo conjunto de soluciones.

Procediendo de esta forma el algoritmo explora el espacio de soluciones generando muestras similares a las existentes entre los mejores candidatos encontrados con la esperanza de generar la solución óptima durante su funcionamiento.

Su naturaleza heurística y la complejidad del problema hacen que la solución encontrada sea un óptimo local del problema.

## 4 Diseño de la red Neuronal

Como ya se ha explicado previamente, este estudio aborda el diseño de un controlador de trayectorias de búsqueda en tiempo mínimo (MTS) a partir de los datos obtenidos por un algoritmo heurístico existente que aprovecha la rápida capacidad de respuesta y bajos requisitos de memoria que aportan las redes neuronales.

En este capítulo se presenta el proceso seguido durante el diseño y entrenamiento de la red, y el análisis de los resultados obtenidos con la red final.

### 4.1 Diseño de la red

En esta sección se recogen los enfoques principales que se han considerado durante el proceso de diseño, y un estudio de los casos conflictivos de entrenamiento.

#### 4.1.1 Enfoques analizados

El primer punto a definir es la entrada de datos a la red ya que esta condicionará notablemente el resto de sus parámetros. Debido a la complejidad del problema, no solo será necesario definir la forma en que éstos se introducen sino también su cantidad.

Los datos de entrada del controlador en este problema son la posición del agente y el mapa de probabilidad para encontrar el objeto. Con el fin de generalizar al máximo los cálculos de la red, todas las probabilidades se referenciarán al agente de tal modo que la red sólo tenga acceso a las posiciones relativas agente-mapa de distribución de probabilidad.

En una primera aproximación, puede pensarse que el modo correcto de presentar los datos es mediante tríos con la estructura representada en la Figura 13.



Figura 13: Datos correspondientes a una celda en la primera aproximación a la red

De esta forma el número de entradas a la red será igual a  $N_R \times 3$ , donde  $N_R$  es el número de celdas de las que tenemos información relevante.

Celdas relevantes pueden ser aquellas con mayores probabilidades, las más cercanas, las que se encuentran dentro del radio de cálculo de la red o simplemente todas las que están dentro de la visión del agente. A continuación analizamos las implicaciones asociadas a los diferentes criterios que puedan convertir a una celda en relevante.

- Celdas con mayores probabilidades. Este criterio queda descartado ya que el objetivo del controlador a diseñar es la búsqueda en tiempo mínimo, que necesita los datos de todas las probabilidades del grid. En la Figura 14 se ilustra este hecho, donde las probabilidades más relevantes están en rojo, seguidas en importancia por las naranjas y las amarillas, y finalmente por las verdes que deberían ser ignoradas por el agente. Si únicamente utilizamos las celdas con mayor probabilidad, podríamos escoger dos caminos diferentes con el mismo resultado, pero que en realidad, tienen diferentes valores de LET.

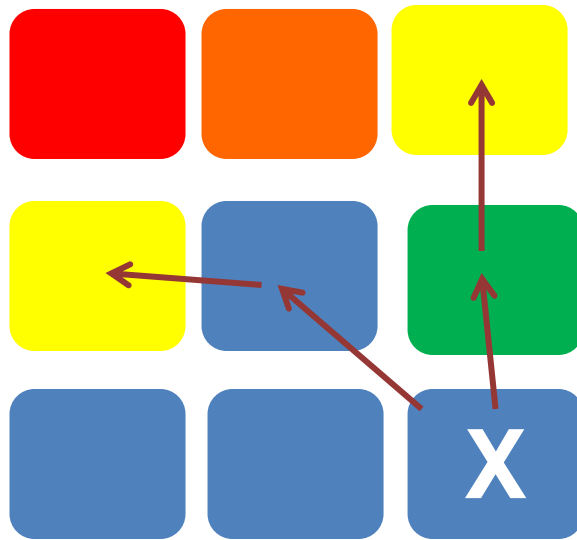


Figura 14: Posibles caminos calculados ignorando las probabilidades más bajas

- Celdas más cercanas. Si el radio de datos de entrada es superior al radio de cálculo del agente, esta solución va a generar una red ineficiente en los casos en los que la probabilidad esté distribuida en toda la circunferencia de acción del agente. En la Figura 15 se puede ver esta situación, la zona sombreada representa el área de la que el agente tiene datos. La trayectoria decidida puede ser la marcada con flechas, y como vemos ésta no es la más conveniente ya que ignora por desconocimiento las probabilidades altas de la primera fila.

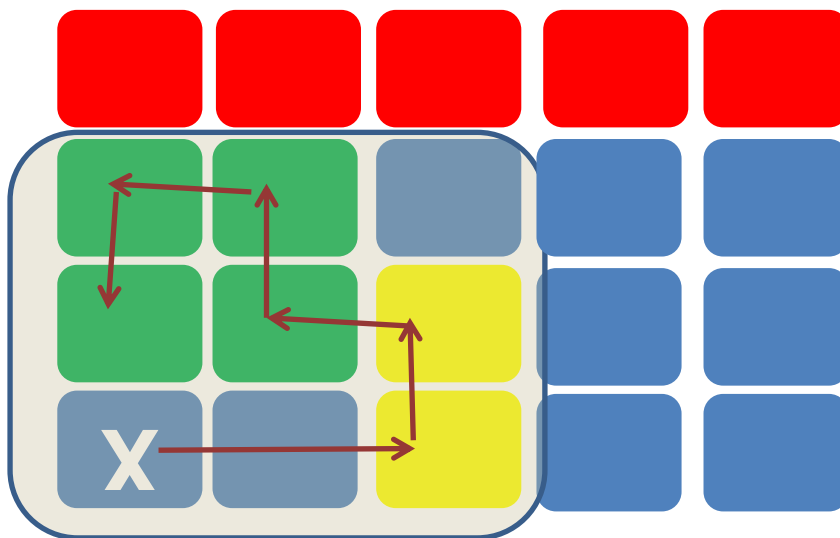


Figura 15: Posible camino calculado con un campo de visión reducido



- Celdas dentro del radio de acción del agente. Esta es la opción que se escoge para los datos de entrada al controlador. Dado que el controlador va a calcular N movimientos, tras los cuales volverá a hacer otro cálculo en función del nuevo mapa de probabilidad, nuestro mundo debe ceñirse a los movimientos que podemos afrontar.

En definitiva, el número de celdas relevante está en función del número de movimientos a predecir. En la Tabla 1 se presenta el número de celdas relevantes para distintos valores del número de movimientos a predecir.

NUMERO DE MOVIMIENTOS	CELDAS TOTALES
1	8
2	24
3	48
4	80
5	120
6	168
7	224
8	288
9	360
10	440

Tabla 1: Número de celdas necesario en función del número de movimientos

En el caso del controlador que nos ocupa, el objetivo es diseñar una red de predicción de N=10 movimientos.

Como resumen de esta primera aproximación al problema tenemos:

- Datos en tríos distancia-ángulo-probabilidad
- 440 celdas relevantes  $\Rightarrow 3 \cdot 440 = 1320$  datos de entrada por cada ejemplo
- Predicción de N=10 movimientos

Esta aproximación tiene varios inconvenientes:

- La gran cantidad de entradas puede hacer lento el proceso de cálculo de cada iteración en la fase de aprendizaje. También puede provocar desbordamientos de memoria en algunos métodos de entrenamiento.
- Las diferentes unidades de las variables de entrada (metros, radianes y probabilidades) pueden provocar problemas en la red para asignar pesos.

Las pruebas que se realizan sobre esta configuración corroboran los inconvenientes anteriores ya que para el entrenamiento Levenberg–Marquardt se produce un desbordamiento de memoria y para los métodos de gradiente conjugado y descendente la red alcanza el mínimo local de 1,65 que se puede ver en la gráfica de entrenamiento de la Figura 16, siendo este resultado insuficiente para los objetivos del controlador.

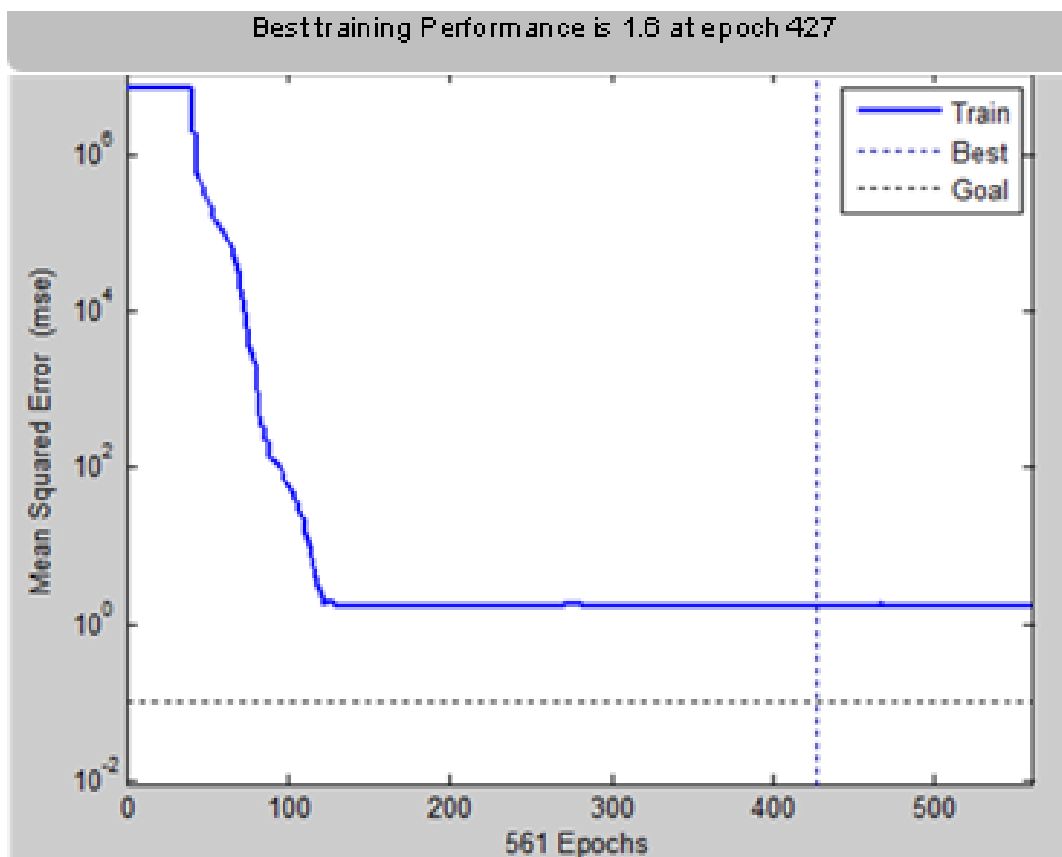


Figura 16: Gráfica de entrenamiento de la red neuronal primera configuración

Descartado el primer enfoque, se hace patente la necesidad de simplificar los datos de entrada en dos frentes: reducción de su número, y simplificación del tipo de unidades. Para hacer esto, hallemos el paralelismo con una red más simple, por ejemplo, un sistema de reconocimiento de caracteres, que se basa en ciertas propiedades (Números de Hu, excentricidad, etc). La red aprende con un conjunto de propiedades y un valor de salida.

Las propiedades en nuestra red pueden ser los valores de probabilidad en las celdas en torno al agente organizadas y numeradas tal y como se muestra en la Figura 17. Además, con el objeto de generalizar al máximo el proceso de aprendizaje, la numeración se orienta hacia la posición de la celda de máxima probabilidad.

15	14	13	12	11
16	4	3	2	10
17	5	X	1	9
18	6	7	8	24
19	20	21	22	23

Figura 17: Numeración de las cuadrículas adyacentes al agente

En este caso, al igual que en el primer enfoque, estableceremos que las celdas significativas son las que se encuentran dentro del radio de cálculo del controlador.

Para este segundo enfoque tenemos:

- Datos de probabilidad organizados según la posición respecto al agente y orientación de máxima probabilidad.
- 440 celdas significativas, lo que se traduce en 440 entradas a la red.
- Predicción de N=10 movimientos.

Para el entrenamiento de la red se barajan las siguientes opciones:

- Método de Levenberg–Marquardt.
- Métodos de gradiente conjugado.
  - Traincgb, traincgf, traincgp.
- Métodos de gradiente descendente.
  - Trainsgd, Traingdm, Traingsx, Traingda.

Al igual que en la primera aproximación, el método de Levenberg–Marquardt no es válido porque la memoria se desborda. Tras probar los diferentes métodos

observamos que el método de entrenamiento de gradiente descendente con retropropagación del momento (Traingdm) es el único viable para este enfoque ya que los otros alcanzan mínimos locales que les impiden obtener un buen rendimiento.

Además, dada la complejidad del problema es necesario introducir una capa oculta que favorezca la convergencia. El esquema de la red neuronal resultante se muestra en la Figura 18. El número de neuronas de las capas de entrada, salida e intermedia es 440, 40 y 10 respectivamente.



Figura 18: Esquema de la red neuronal diseñada

La Figura 19 muestra la evolución del entrenamiento a lo largo de 250.000 iteraciones para un conjunto de 400 ejemplos. En ella se comprueba que este modelo efectivamente converge y reproduce satisfactoriamente los datos de entrenamiento. Cabe destacar que este hecho no es condición suficiente pero sí necesaria para validar este enfoque.

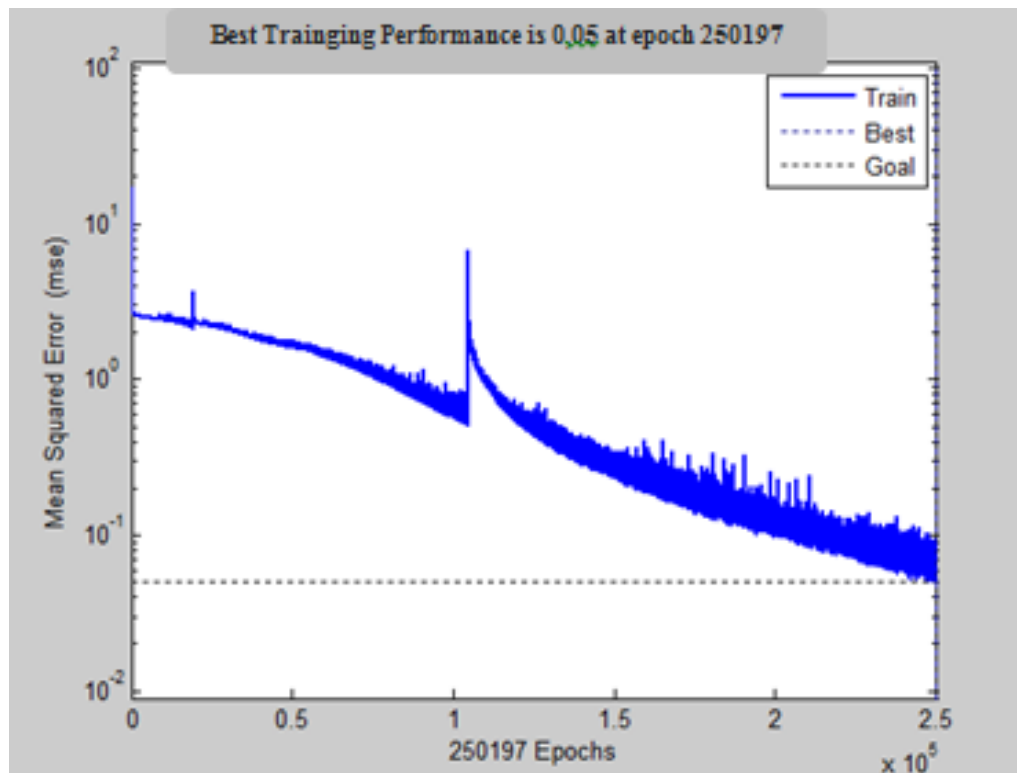


Figura 19: Gráfica de entrenamiento para la aproximación número 2

Los resultados del procesamiento evaluados sobre el conjunto de pruebas que se empleó para el entrenamiento de la red en esta segunda aproximación se muestran en la Figura 20. Los resultados son adecuados ya que un 88% de los casos tiene todas las secuencias de control coincidentes con las calculadas en la heurística. A pesar de que los resultados que realmente necesitamos para validar el funcionamiento de la red son los de un conjunto de test que no tiene relación con el de entrenamiento, estos buenos resultados muestran que la red está aprendiendo los casos que pertenecen al conjunto de entrenamiento, hecho que supone un buen punto de partida.

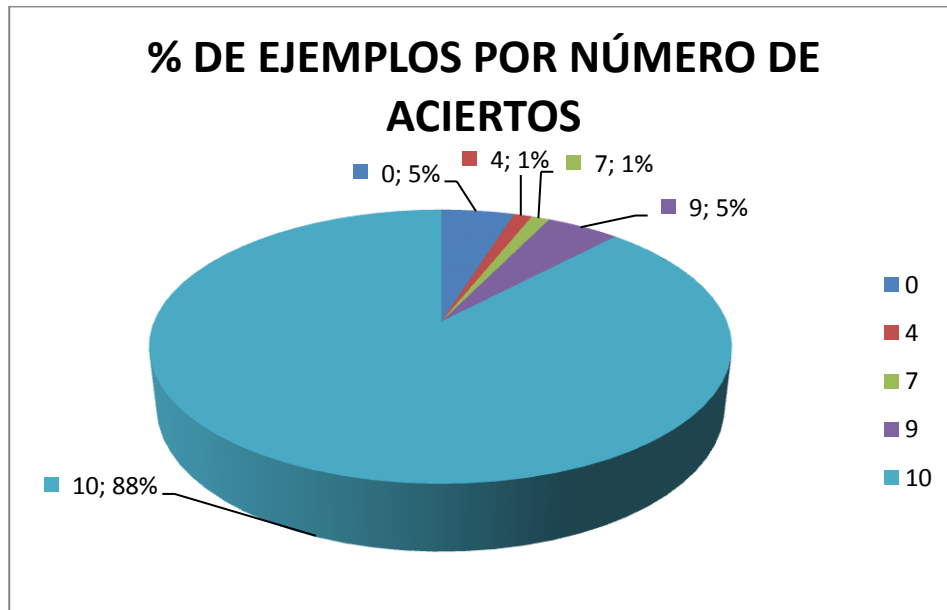


Figura 20: Resultados del procesamiento del conjunto de entrenamiento de prueba mediante la red neuronal obtenida en la aproximación 2

#### 4.1.2

#### 4.1.3 Estudio de los datos conflictivos del entrenamiento

Antes de entrar en mayor detalle con los aspectos de ajuste y validación de la red neuronal se hace necesario un estudio previo de los casos de entrenamiento que va a aprender la red.

Los problemas con los que va a trabajar la red están agrupados en tres tipos: distribución simétrica de dos gaussianas, no simétrica de varias gaussianas y aleatoria. Las probabilidades se encuentran ubicadas en un grid de 20 x 20 celdas. El agente puede estar situado en cualquier posición de esta cuadrícula incluyendo los casos en los que se localiza en la zona con mayor densidad de probabilidad o en la esquina opuesta.

Las características de algunos ejemplos que requieren cuidado especial por poner en riesgo la convergencia de la red son las siguientes:

- a) Distribuciones simétricas: la simetría en las distribuciones de probabilidad implica la existencia de varios caminos igualmente buenos. Un ejemplo se puede ver en la Figura 21, si tenemos una distribución simétrica y el agente está situado hacia la mitad de la distribución puede tomar dos caminos de direcciones opuestas. Lo que podría parecer ventajoso, ya que si hay mayor número de trayectorias con buen resultado mayores son las probabilidades de encontrar un camino bueno, supone un inconveniente ya que la inclusión en el conjunto de entrenamiento de dos ejemplos con distribución parecida y caminos muy diferentes puede confundir a la red. Es decir, si introducimos a la red neuronal dos ejemplos con esta distribución que toman direcciones opuestas, esto supondrá un atraso en el aprendizaje ya que la secuencia aprendida para un ejemplo es opuesta para el otro.
- b) Falta de datos significativos en torno al agente: esto es debido a la lejanía del agente respecto a la zona donde se concentra la probabilidad. No hay que olvidar que el agente se sitúa en una posición aleatoria dentro de una cuadrícula de 20x20, tiene un radio de alcance de 10 pasos, y la probabilidad de encontrar el objeto puede estar concentrada o distribuida sobre cualquier región del espacio. Por lo tanto se pueden generar algunos casos extremos en los que el agente no puede acumular ninguna probabilidad dentro de su campo de acción. En la figura 22 se puede ver esquemáticamente un caso en el que el agente no va a poder alcanzar ninguna casilla que tenga probabilidad mayor que cero (naranjas). La inclusión de varios ejemplos de entrenamiento con una configuración similar también supone un riesgo para la convergencia de la red, por ejemplo, si el agente no ve ninguna probabilidad el camino calculado por la heurística es aleatorio ya que cualquier camino que calcule va a recoger cero probabilidades.



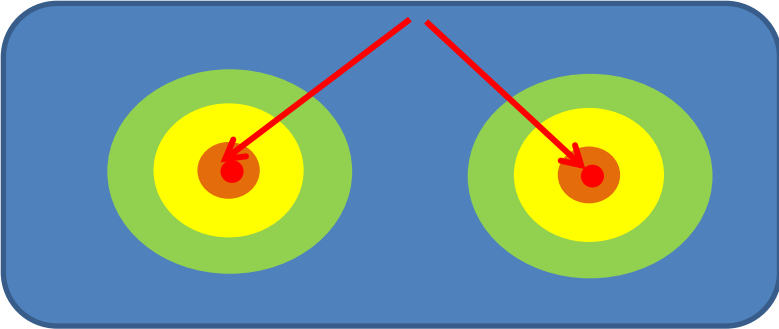


Figura 21: Caminos posibles en una distribución simétrica

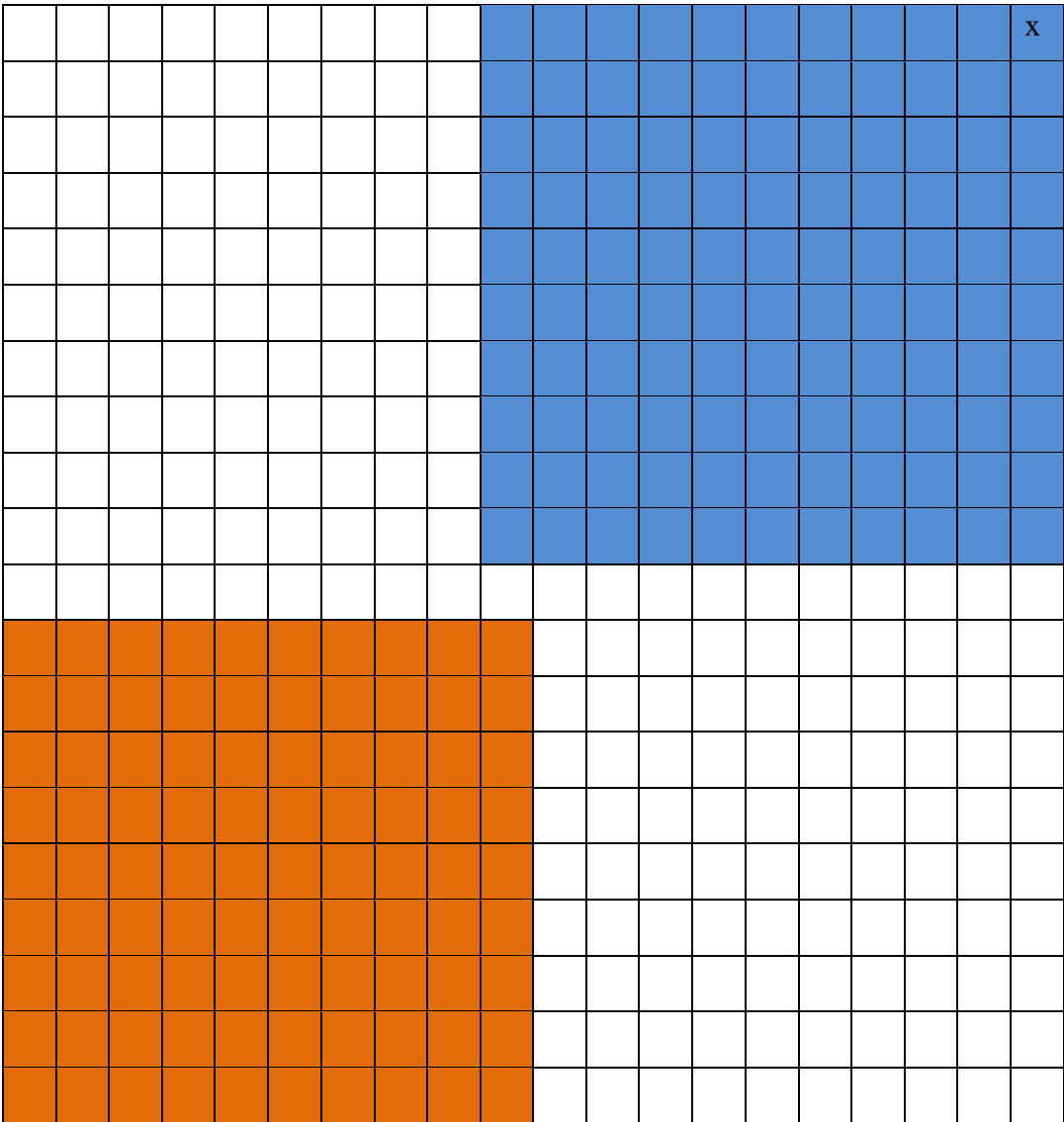


Figura 22: Distribución de creencias fuera del campo de visión del agente

Tal y como ya se introdujo en el punto “a)”, las distribuciones simétricas son un supuesto que resulta delicado porque existen varias trayectorias con valores de bondad muy similares. Si se introducen distribuciones simétricas en el conjunto de entrenamiento se crea el riesgo de introducir varios ejemplos muy parecidos entre sí con trayectorias de búsqueda opuestas. Si se cumpliera lo anterior, la red aprendería la secuencia de salida para uno de los casos, pero esto se revertiría al aparecer otro ejemplo de distribución similar y trayectoria de salida diferente. Esta situación se puede dar también en casos que no son simétricos aunque las probabilidades de encontrar varios ejemplos con distribución similar y trayectorias diferentes son muy bajas, además las diferencias en las trayectorias no serán en ningún caso tan marcadas como las que existen en distribuciones simétricas. El entrenamiento para este tipo de casos requiere un filtrado previo de los ejemplos, teniendo en cuenta que es necesario contemplar toda la casuística de distancias y radios de distribución. En este estudio este filtrado se ha realizado de forma manual.

Como se introduce en el punto “b)”, el campo de visión del agente también ha resultado ser un importante limitante a la hora de ajustar la red, no hay que olvidar que el agente no tiene por qué estar centrado en los mapas de creencia de los casos (dimensiones 20x20) ya que puede ocupar cualquier celda del grid. Con esto, tenemos que el agente puede estar tan alejado de la distribución que no alcanzará nunca ninguna creencia aunque siga el camino correcto. Es más, como sólo se aportan los datos de las casillas que están dentro del radio de acción del agente, las áreas con mayor probabilidad no se incluirán en los datos de entrada a la red. Para resolver este problema, también se aplica un filtro para que no se tengan en cuenta los casos con menos de un décimo de las probabilidades del mapa en el campo de actuación del agente. A modo de ilustración, en las figuras 23, 24 y 25, están representadas tres situaciones que la red entendería como la misma. El cálculo de trayectorias en estos casos no es útil ya que la red no está teniendo en cuenta la mayor parte de las probabilidades.

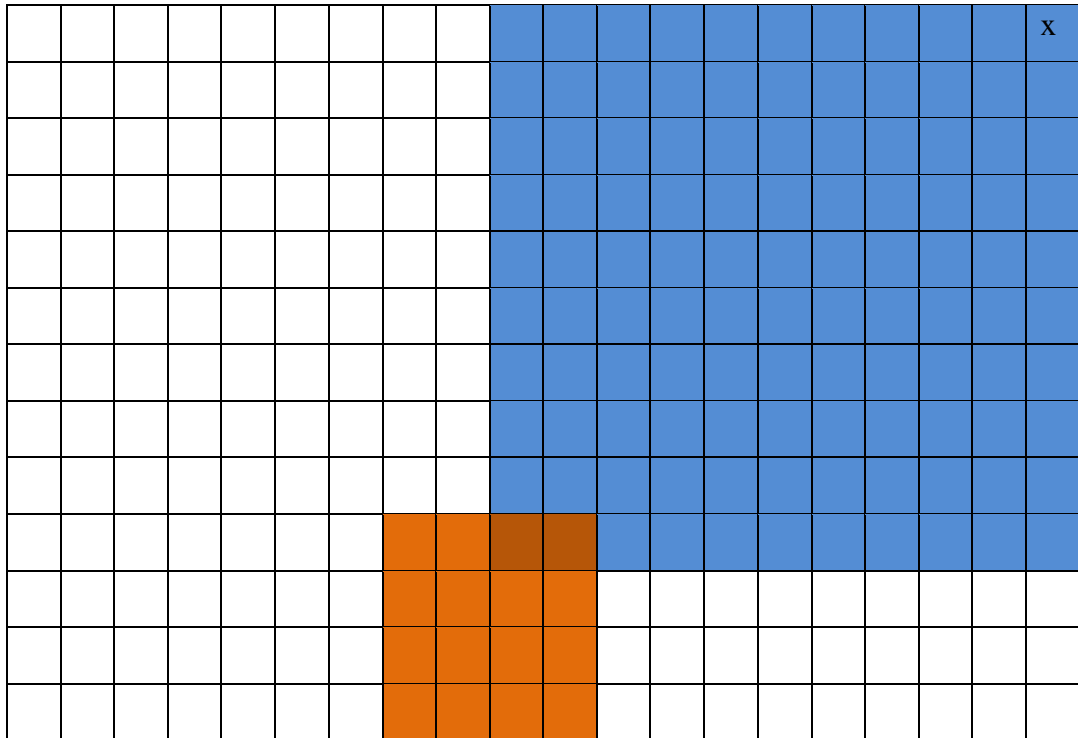


Figura 23: Situación en la que menos de una décima parte de las probabilidades están dentro del campo de visión

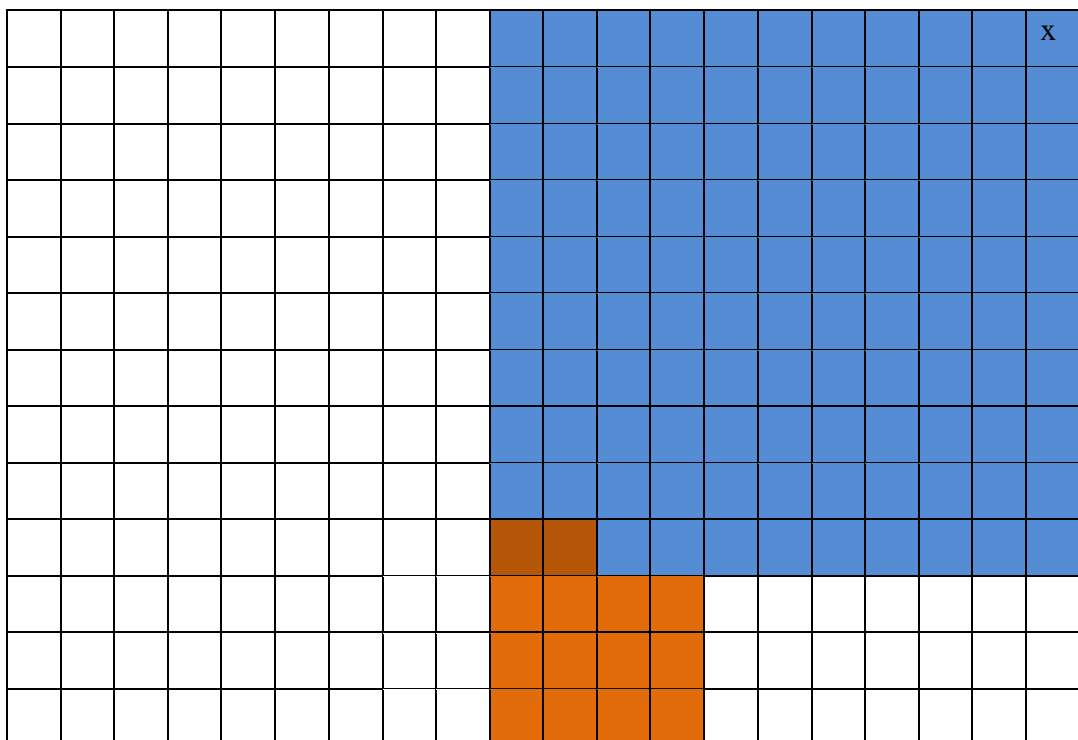


Figura 24: Situación en la que menos de una décima parte de las probabilidades están dentro del campo de visión del agente

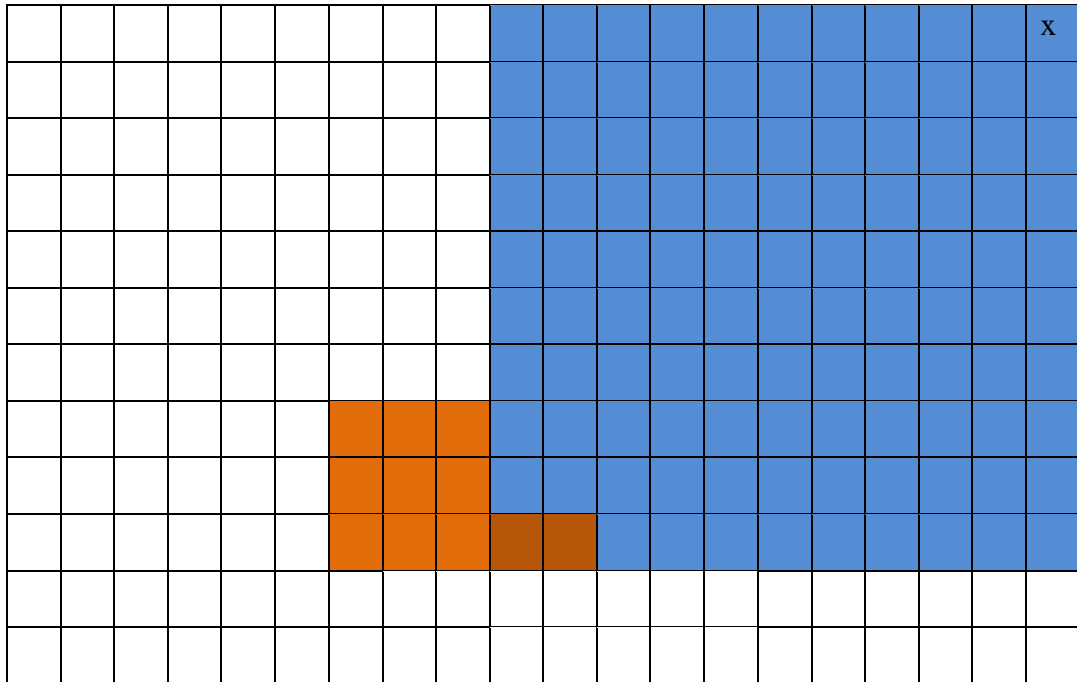


Figura 25: Distribución de probabilidad en la que menos de una décima parte de las probabilidades se encuentra dentro del campo de visión del agente

## 4.2 Entrenamiento de la red

En esta sección se presenta la métrica utilizada durante el proceso de entrenamiento, y las etapas y resultados obtenidos durante el mismo.

### 4.2.1 Medida de calidad de los resultados

Una vez definidas las características generales de la red neuronal, es necesario establecer una medida de la calidad de los resultados para ser capaces de evaluar los progresos en el ajuste.

En el apartado 4.1.1 se ha obtenido una configuración que reproduce a la perfección el 88% de los ejemplos de entrenamiento en los que los casos especiales ya se habían filtrado, pero no se debe olvidar el hecho de que existe un número infinito de ejemplos y de nada sirve intentar buscar una red que reproduzca el 100% de las posiciones porque siempre aparecerán nuevos casos que no podrán cumplir con esta expectativa.

Para determinar la validez de un resultado es necesario tener en cuenta el objetivo que se está buscando: minimizar el tiempo de búsqueda. El grado de consecución del objetivo se mide mediante el parámetro LET explicado en puntos anteriores. Dado que la red origina 10 salidas por cada ejemplo, la variable LET se calculará para 10 pasos siendo el valor de 10 el peor valor que puede obtenerse ya que implica que el agente ha pasado en todo momento por celdas de probabilidad nula.

Con todo lo expuesto podemos considerar dos medidas:

- Número de aciertos.
- Reducción del tiempo de búsqueda estimado respecto al total. Esta medida se obtendrá con la ecuación 1. El signo negativo de la ecuación se debe a que buscamos el valor de la reducción de tiempo que supone el cálculo con la red, por ejemplo, si la trayectoria calculada con la red neuronal tiene LET más pequeño que la calculada con la heurística, la diferencia entre estos tiempos es negativa, pero el ahorro obtenido con la red neuronal es positivo. En resumen, para la métrica definida en la ecuación 1, los mejores valores son los más positivos y los peores los más negativos.

$$\Delta = -[(LET_R - LET_H) / (10 - LET_R)] * 100 \quad (1)$$

Para analizar la fiabilidad de cada una de las métricas propuestas así como la relación existente entre ellas, se procesa un conjunto de casos de prueba al azar, se ordenan mediante el criterio de reducción de tiempo de búsqueda y se representan los resultados de ambas métricas valorados para cada uno de los ejemplos como se muestra en la Figura 26 que tiene reseñados en verde los valores que representan una reducción de LET en las trayectorias calculadas mediante la red neuronal. El valor de corte a partir del cual se considera que los

ejemplos procesados no han tenido un resultado aceptable se establece de forma experimental y se señala en la Figura 26 mediante una línea roja.

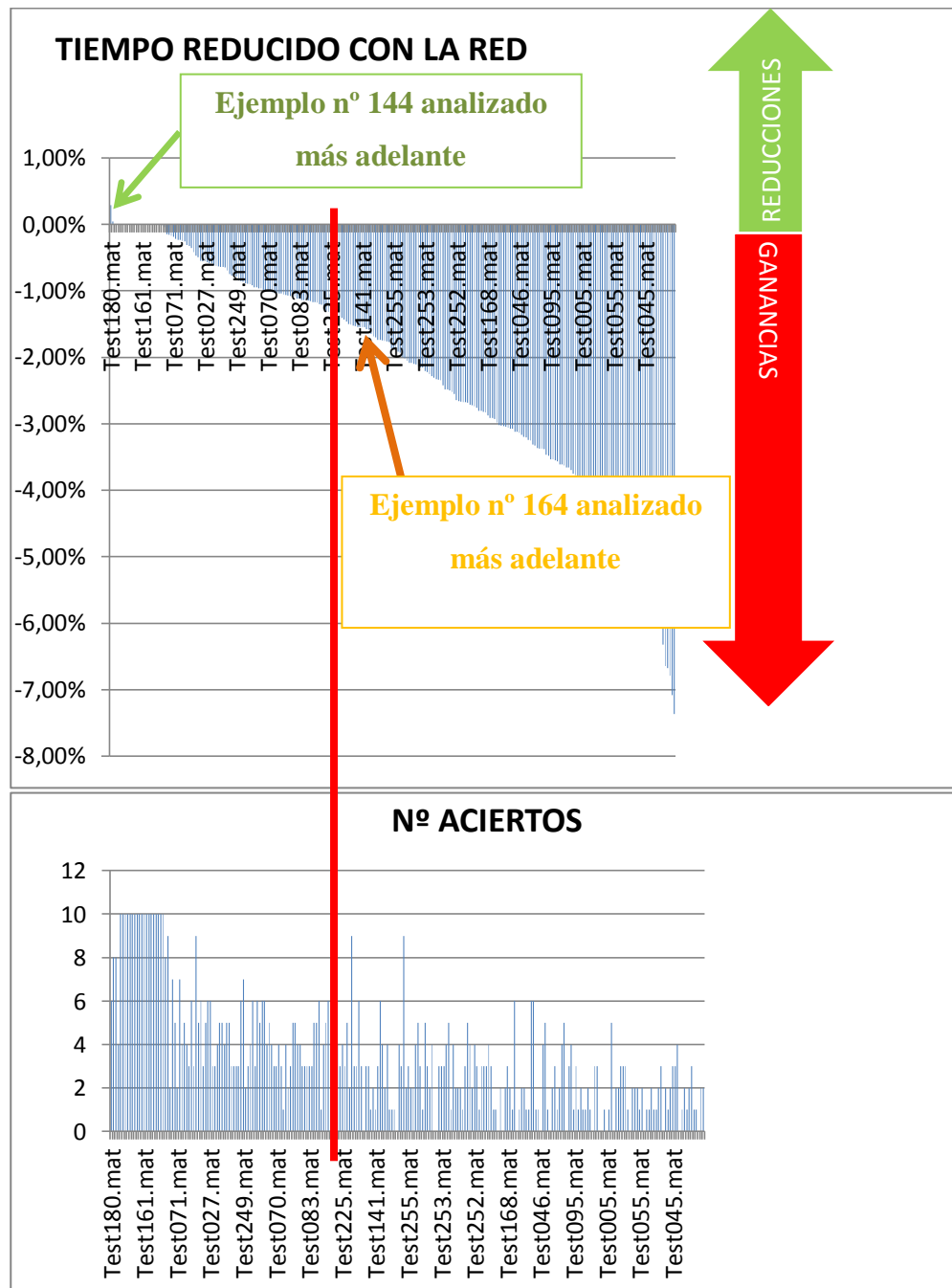


Figura 26: Comparativa de métricas relacionadas con LET y número de aciertos para un conjunto de datos de test

El tiempo mínimo estimado calculado por la red para el ejemplo de prueba 144, señalado en la Figura 26, tiene menor valor que el tiempo mínimo estimado para la trayectoria calculada con la heurística. La red en este caso supone un ahorro del 0,45% del tiempo mínimo estimado. En contraste con esto, la métrica que indica el número de aciertos respecto a la red neuronal tiene valor 4. Si ésta última fuese la métrica elegida para valorar la bondad, este ejemplo de prueba no se aceptaría como bueno a pesar de suponer una reducción en el tiempo mínimo estimado. En las Figuras 26b y 26c se puede apreciar la diferencia entre estas trayectorias.

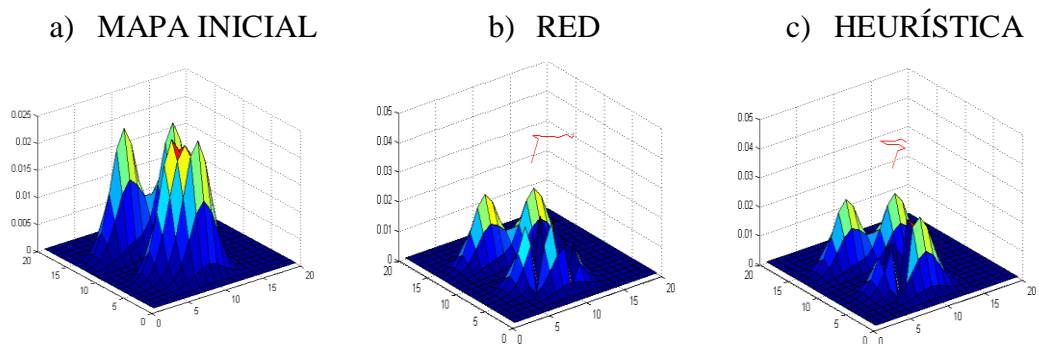


Figura 27: Análisis de ejemplo con mejora en el valor de tiempo mínimo estimado

En contraste con lo sucedido en el ejemplo anterior, para la trayectoria calculada en el caso de prueba 164 representado en la Figura 28, el número de aciertos respecto a la heurística es 6 pero el valor del ahorro de tiempo es -1,92% lo que supone que la trayectoria calculada por la red supone un gasto adicional de tiempo.

Si se analizan las Figuras 28b y 28c, puede verse que la trayectoria obtenida por la red no ha recogido gran parte de las probabilidades recogidas por la trayectoria obtenida con la heurística

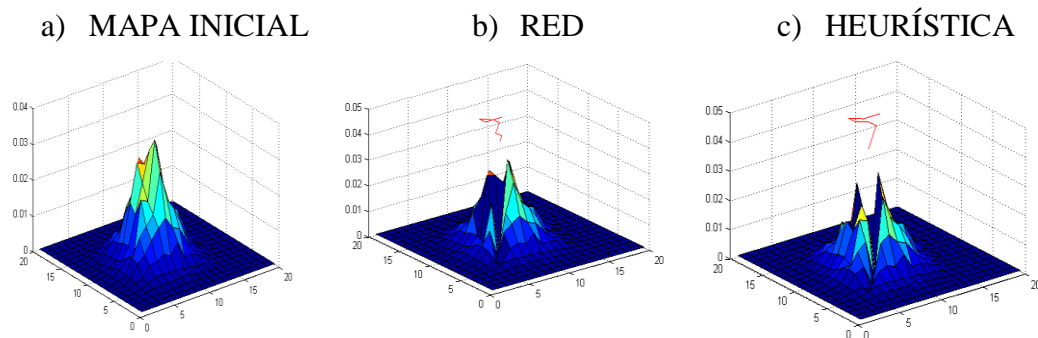


Figura 28: Análisis de caso en el que la red no recoge suficientes probabilidades

Llegados a este punto resulta evidente que la medida de reducción de LET es necesaria, pero cabe preguntarse si tiene alguna relación con el número de aciertos de la red. Para responder a esta pregunta, se ordenan los resultados del procesamiento de un conjunto de test de mejor a peor en base al criterio de mejora de LET y se comparan ambas métricas.

En la Figura 26 se observa sobre un conjunto de 200 ejemplos de prueba que los resultados que suponen una mejora respecto al cálculo heurístico no tienen necesariamente un número de aciertos elevado. En la misma, la línea roja determina la frontera entre los ejemplos válidos y los que no lo son. De hecho se puede observar la existencia de una serie de casos en los que la salida de la red es una reproducción exacta del cálculo de la heurística en los que el valor relacionado con la mejora de LET es cero (ni mejoramos ni empeoramos), y después de estos se aprecia que no existe relación entre el número de aciertos y el valor del porcentaje de mejora.

Como ya se comentó anteriormente, en la Figura 26 se ha representado una línea que separa los casos que entenderemos como aceptables de los que no lo son. Un caso resultará aceptable si la trayectoria calculada por la red mejora la calculada por la heurística en función de la métrica elegida o al menos no la empeora demasiado. El valor de corte que se sitúa en -1,5%, en otras palabras la red



provoca un aumento del tiempo mínimo de 1,5% respecto a la heurística. Este valor se ha estimado de forma experimental con varios ejemplos.

En la Figura 29 se muestran varios ejemplos cuyos valores de  $\Delta$  (métrica de reducción de LET) se encuentran próximos a la frontera. Se puede apreciar cómo las trayectorias de las filas coloreadas en rojo (no aceptables según la frontera de - 1,5%) dejan bastantes probabilidades sin recoger. Sin embargo, las trayectorias de las filas señalizadas en verde recogen mayor cantidad de probabilidad.

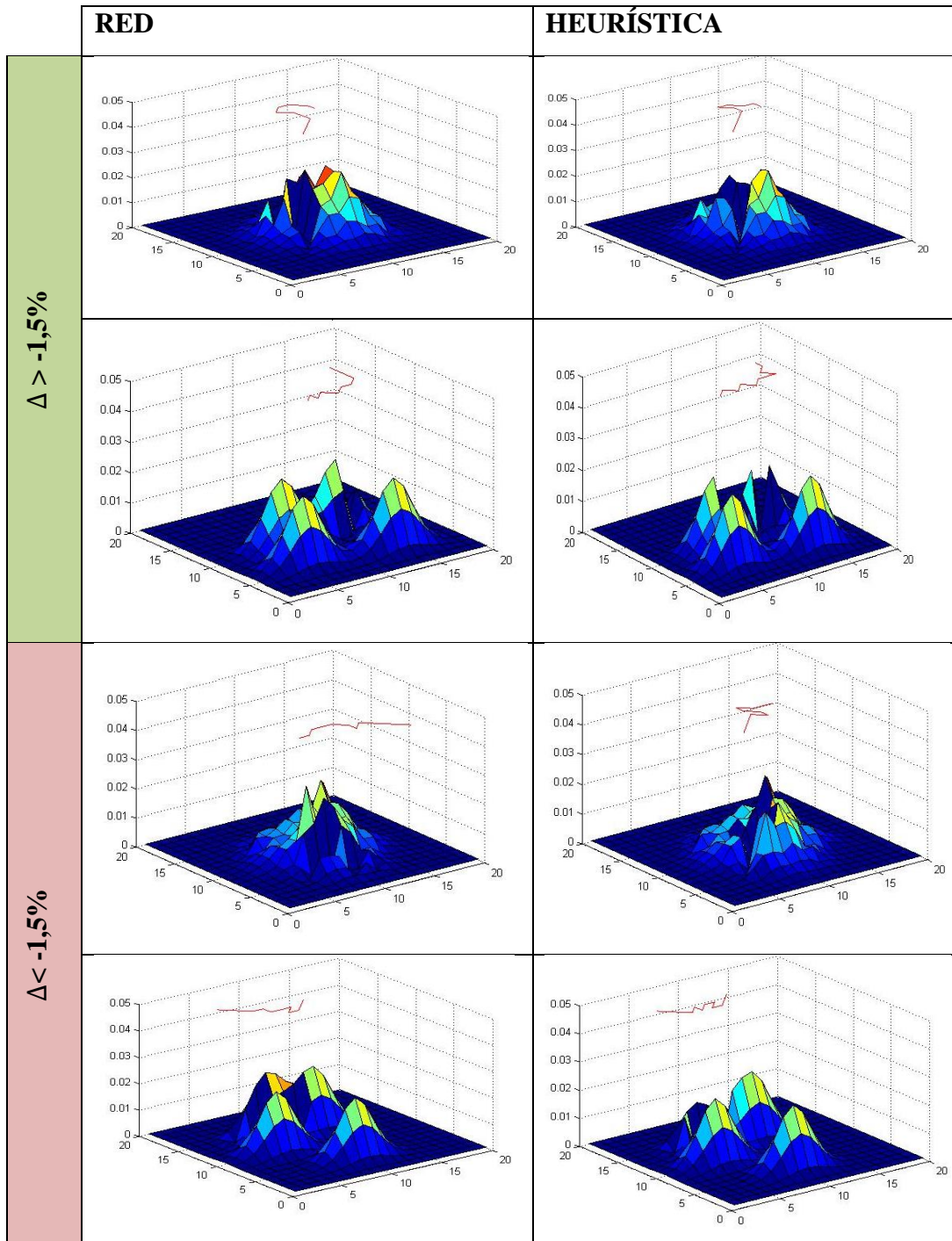


Figura 29: Casos a ambos lados de la frontera de medidas aceptables

Resumiendo todo lo expuesto en este apartado, tenemos que la medida de calidad de los resultados en función de LET ( $\Delta$ , Ecuación 1) es necesaria y sabemos que los resultados cuya medida de calidad es inferior a -1,5% no son aceptables por dejar una importante cantidad de probabilidades sin recoger.

#### 4.2.2 Ajustes y validación de la red

Una vez decididas la configuración de la red neuronal y la métrica utilizada para evaluar la bondad de las soluciones, es necesario definir los objetivos que debe cumplir la red.

En primer lugar se definen los cuatro conjuntos de ejemplos siguientes:

- **CONJUNTO DE PRUEBAS INICIALES:** Para llegar hasta este punto de desarrollo (configuración de red y métricas definidas), se ha empleado un conjunto de 700 ejemplos que no se van a utilizar a partir de aquí.
- **CONJUNTO DE ENTRENAMIENTO:** Está formado por todos los ejemplos que se emplean para entrenar la red. Inicialmente se crea con 400 casos elegidos al azar que se va actualizando con más ejemplos en cada fase de entrenamiento.
- **CONJUNTO DE VALIDACIÓN:** Está formado por una serie de ejemplos escogidos al azar. Nunca se empleará para alimentar el conjunto de entrenamiento.
- **CONJUNTO DE TEST:** Este conjunto es necesario para ratificar la validez del funcionamiento de la red cuando el conjunto de validación ha obtenido buenos resultados. El final de la etapa de entrenamiento está determinada por la obtención de resultados satisfactorios en este conjunto. Al igual que el conjunto de validación nunca se empleará para alimentar el grupo de entrenamiento.
- **CONJUNTO REALIMENTACIÓN DE ENTRENAMIENTO.** Está formado por un conjunto de ejemplos generados al azar y sirve para realimentar el conjunto de entrenamiento. Este conjunto se procesará

mediante un filtro muy básico para evitar que se introduzcan entre los ejemplos de entrenamiento nuevos casos para los que la red tiene buena respuesta. Los casos que se transfieren al entrenamiento son los que han tenido peores resultados al procesarse con la red. Estas operaciones suponen un filtrado muy básico porque solamente aseguran que los nuevos ejemplos de entrenamiento no están aprendidos y por lo tanto, no sabemos si existen casos con distribuciones parecidas entre los que transferimos.

Para entrenar la red neuronal se va a seguir el proceso que se muestra en la Figura 30.

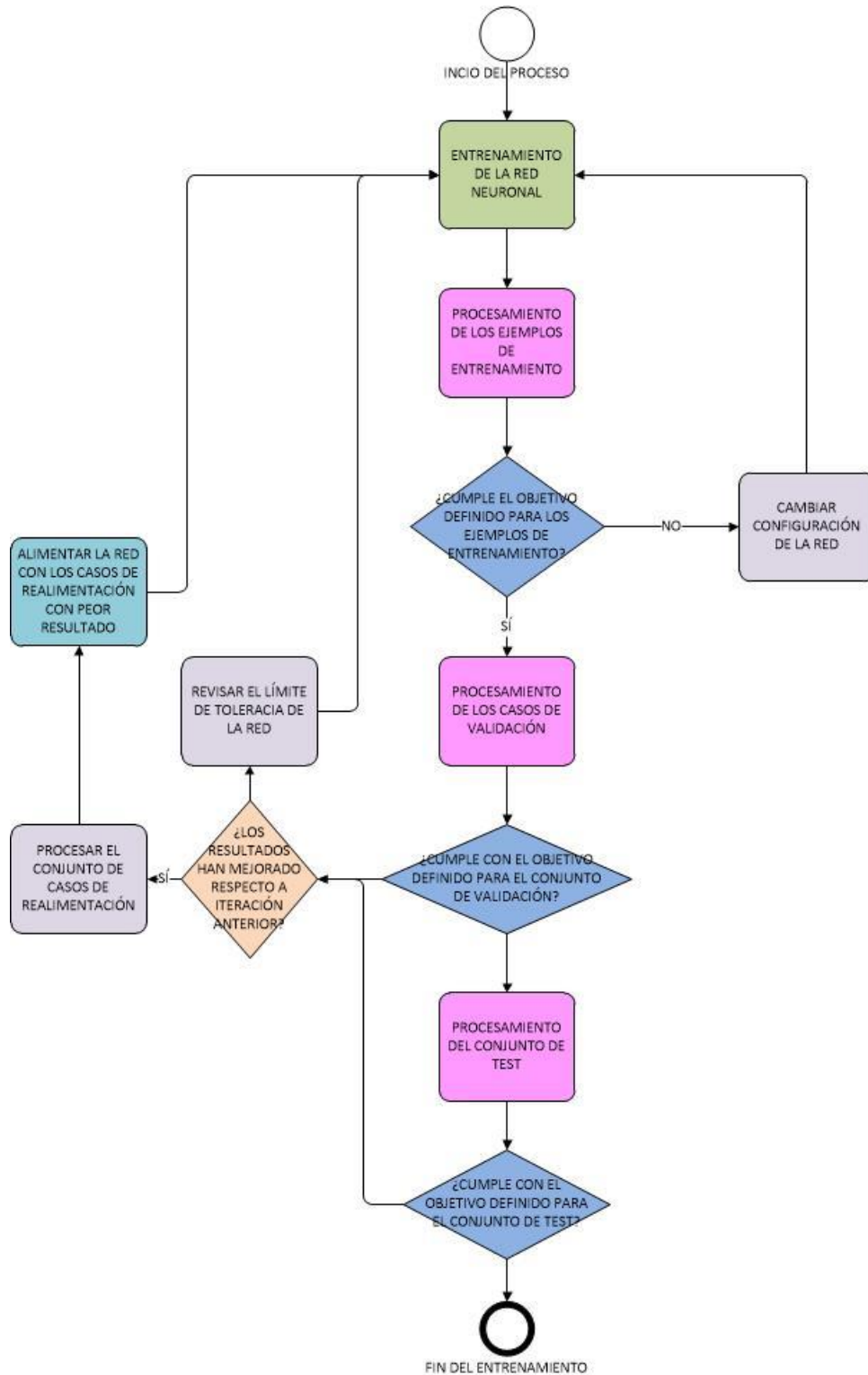


Figura 30: Proceso de ajuste de la red neuronal

Los objetivos que se establecen para este entrenamiento son:

1. CONJUNTO ENTRENAMIENTO. La red debe obtener resultados aceptables para al menos el 80% de las muestras de este conjunto
2. CONJUNTO VALIDACIÓN. La cantidad de los resultados aceptables del procesamiento de este conjunto como mucho debe ser un 15% inferior al número de resultados aceptables del conjunto de entrenamiento
3. CONJUNTO DE TEST. Tiene que obtener unos resultados parecidos a los obtenidos para el conjunto de validación.

Inicialmente, el parámetro de rendimiento del proceso de aprendizaje se fija en 0.05.

Tal y como se ve en la Figura 31a, el porcentaje de ejemplos descartados para el conjunto de entrenamiento es del 8%. Para cumplir los objetivos marcados para la red, este valor debe ser inferior al 20%. Como la red inicial cumple con los requisitos necesarios para el conjunto de entrenamiento, se procesa el conjunto de validación que es el siguiente que debe cumplir con estos objetivos según la Figura 30.

Para representar la Figura 31b se toman por aceptables todos los ejemplos cuya medida de calidad (reducción de tiempo LET respecto a la heurística) sea mayor o igual a -1,5%, y se marcan como descartados los casos en los que no cumple.

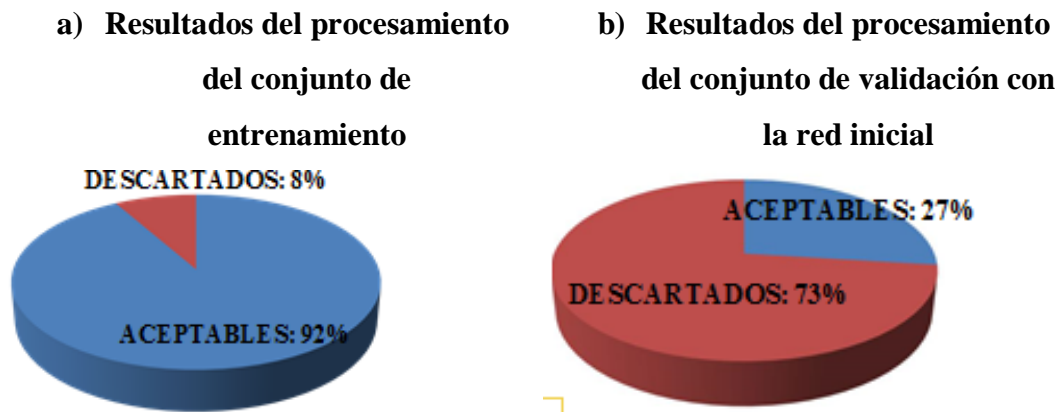


Figura 31: Resultados del procesamiento con la red inicial e los conjuntos de test y entrenamiento

Como los resultados del procesamiento del conjunto de validación no cumplen con los objetivos definidos al principio de este punto, tenemos que ajustar la red siguiendo el proceso que se muestra en la Figura 30. La evolución de los resultados obtenidos para el conjunto de validación a lo largo de las diferentes actualizaciones de la red se muestra en la Figura 32.

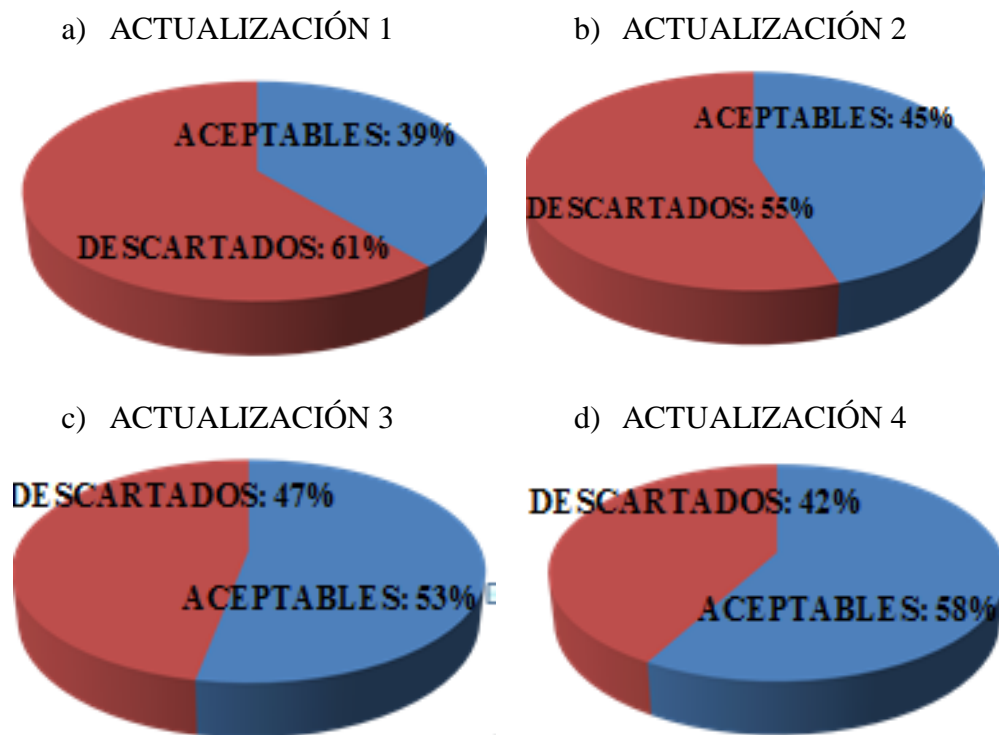


Figura 32: Evolución de los resultados en el conjunto de validación a lo largo del desarrollo de la red

Hasta la actualización número 4, los resultados evaluados sobre el conjunto de validación mejoraron en cada iteración, por lo que hasta este momento lo único que se hizo es realimentar el grupo de casos de entrenamiento con los ejemplos definidos para tal fin en el “conjunto de casos de realimentación de entrenamiento”. Las actualizaciones sucesivas no ocasionaron mejoras en los resultados por lo que se ajustó el parámetro del rendimiento del proceso de aprendizaje cambiando su valor anterior de 0,05 a 0,4.

En la Figura 33a se puede ver que el procesamiento del conjunto de entrenamiento por la red actualizada tiene un 17% de ejemplos descartados. Aunque estos resultados son peores si los comparamos con los que teníamos antes de modificar el rendimiento de la red aunque siguen cumpliendo con los objetivos marcados para la red, si atendemos a la Figura 33b, los resultados para el procesamiento del conjunto de validación han mejorado de un 58% de casos aceptables que se tenían en la actualización número cuatro hasta el 75% obtenido después de modificar el rendimiento del proceso de aprendizaje de la red.

La mejoría obtenida en esta actualización se debe a que la red con la que estábamos trabajando inicialmente estaba sobre-ajustada y sólo calculaba de forma aceptable los ejemplos con un elevado parecido a los de entrenamiento. La red que se ha obtenido en la actualización 5 cumple con el requisito básico de aprendizaje de una red neuronal: aproximar una función no lineal que cumpla con los ejemplos de entrenamiento y validación.

- a) Resultado del procesamiento del      b) Resultado del procesamiento



conjunto de entrenamiento  
después de modificar el  
parámetro de rendimiento de la  
red (actualización 5)

del conjunto de validación  
después de modificar el  
parámetro de rendimiento de la  
red (actualización 5)

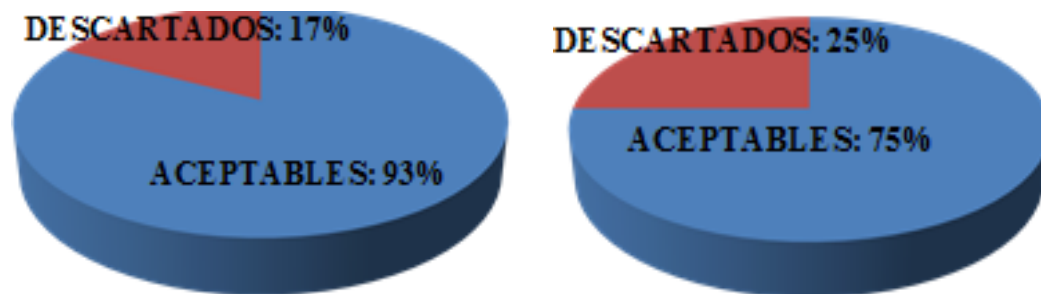
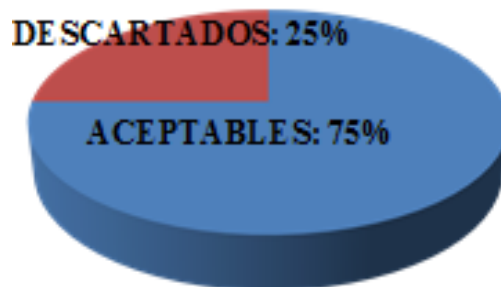


Figura 33: Resultados del procesamiento de los conjunto de entrenamiento y validación después de la modificación del parámetro "performance" (actualización 5)

Como el porcentaje de casos descartados en el conjunto de entrenamiento es 17%, el máximo de ejemplos descartados que se pueden aceptar para cumplir con los objetivos es 32% (un 15% superior al porcentaje de descartados en el conjunto de entrenamiento). Como el porcentaje de ejemplos descartados es 25% la red cumple con los objetivos definidos para el conjunto de validación. Por lo tanto, el siguiente paso es comprobar los resultados de la red sobre el conjunto de test.

La evaluación de las trayectorias calculadas sobre el conjunto de test se muestra en la Figura34. Como el requisito que ha de cumplir el conjunto de test es el mismo que para el de validación, (un 15% superior al porcentaje de descartados en el conjunto de entrenamiento), la red obtenida cumple con todos los objetivos definidos para ella.



**Figura 34:** Resultado del procesamiento del conjunto de test con la red neuronal actualizada

Tras realizar este proceso, se han cumplido los tres objetivos establecidos para el ajuste de la red:

1. **CONJUNTO ENTRENAMIENTO.** La red debe obtener resultados aceptables para al menos el 80% de las muestras de este conjunto. Los resultados son aceptables en el conjunto de entrenamiento ya que después del procesamiento representan el 83%.
2. **CONJUNTO VALIDACIÓN.** La cantidad de los resultados aceptables del procesamiento de este conjunto como mucho debe ser un 15% inferior al número de resultados aceptables del conjunto de entrenamiento. Con los resultados obtenidos para el conjunto de entrenamiento, los resultados del procesamiento del conjunto de validación deben representar un porcentaje de casos descartados inferior al 32%. El conjunto de validación tiene un 25% de casos descartados, cumpliéndose el requisito sobre este conjunto.
3. **CONJUNTO DE TEST.** Tiene que obtener unos resultados parecidos a los obtenidos para el conjunto de validación. Los resultados del procesamiento del conjunto de test son los mismos que para el conjunto de validación.

Por lo tanto, tomamos la red obtenida en la actualización 5 (última llevada a cabo) como definitiva y congruente con los objetivos marcados.

Una vez aceptada la red obtenida por cumplir con los objetivos, es necesario analizar el conjunto de entrenamiento ya que la bondad de las soluciones

procesadas por la última actualización de la red empeoró, situándose en 17% el porcentaje de descartados.

Si atendemos a la Figura 35, donde se han ordenado los resultados de mejor a peor, podemos ver que los casos que han obtenido peores resultados tienen un número de aciertos mayor a 5, de hecho, los peores resultados tienen 8 aciertos.

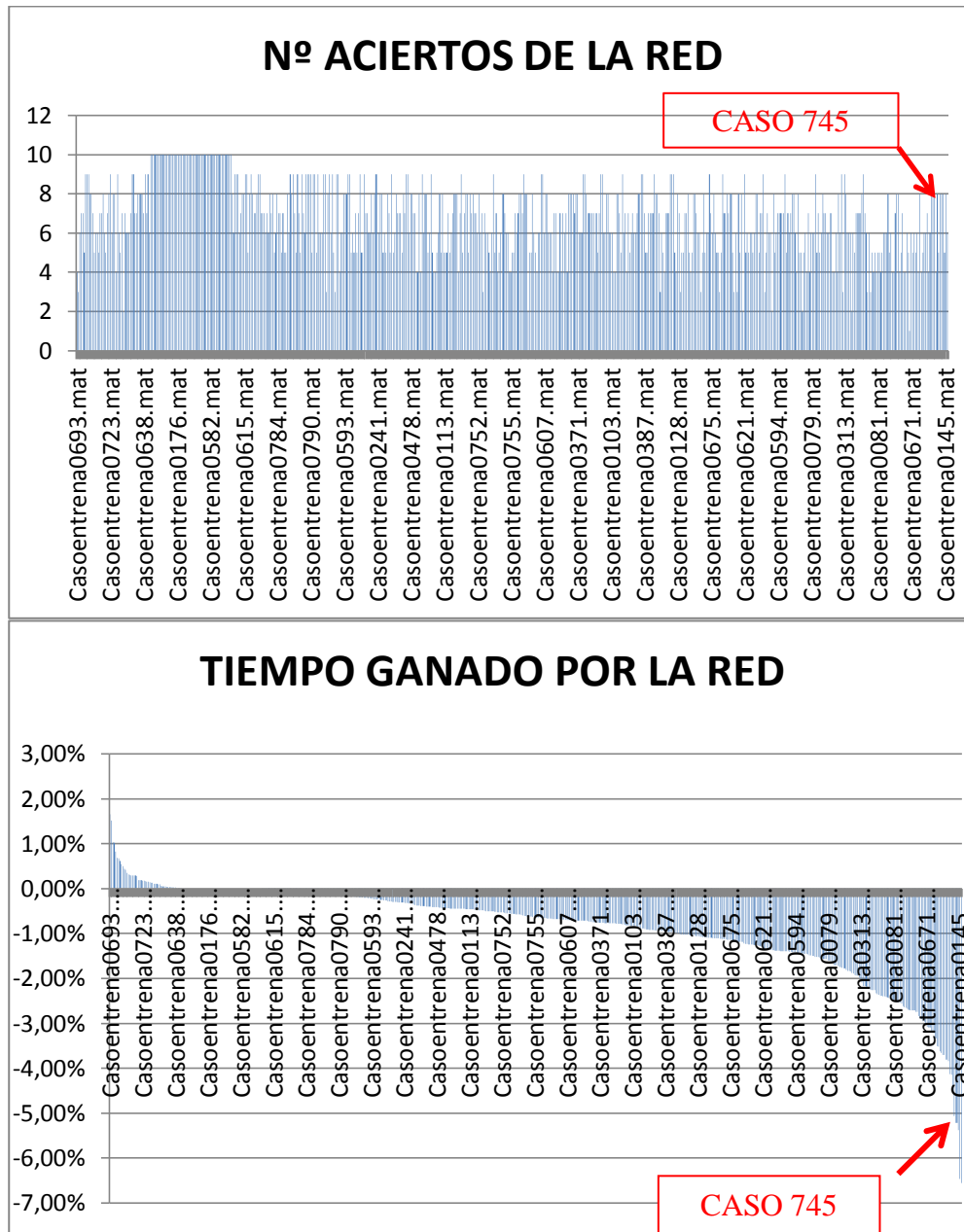


Figura 35: Representación de los resultados obtenidos por la red final para el conjunto de entrenamiento

En la Figura 36 se muestra uno de los casos con peor resultado al procesarse con la red, con una coincidencia de 8 posiciones respecto a la trayectoria heurística. Este ejemplo se encuentra reseñado en la Figura 35 como caso de entrenamiento 745.

En el ejemplo, sólo se modifican dos de las secuencias de control de la trayectoria del agente, pero son suficientes para que éste no recoja algunas probabilidades con valor elevado. Esto sucede porque las probabilidades están muy concentradas en determinadas zonas del espacio y una desviación de estas áreas por ligera que sea provoca un aumento importante en el valor de LET.

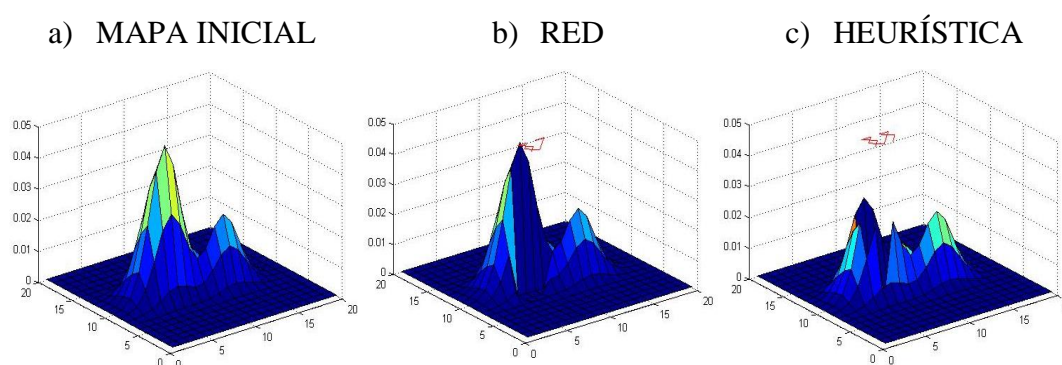


Figura 36: Caso de entrenamiento reproducido de forma no eficiente por la red neuronal

### 4.3 Resultados

En esta sección se analiza el funcionamiento de la red fina, recogiendo un estudio de los resultados obtenidos sobre el conjunto de test, y una comparativa entre los resultados obtenidos con la red y con el algoritmo de optimización utilizado para generar los datos de entrenamiento.

#### 4.3.1 Análisis de los resultados del conjunto de test

En este punto se analizan los resultados obtenidos al procesar el conjunto de test mediante la red obtenida. Se ha elegido este conjunto ya que es el último conjunto que se tiene que evaluar para comprobar el cumplimiento de la red según la Figura 30.

En la Figura 37 se muestra la comparativa “resultados de la red vs nº de aciertos”, donde los casos descartados, que se pueden ver sombreados en rojo, (reducción del tiempo inferior al -1,5%) tienen un número de aciertos respecto a la heurística

que oscila desde 1 hasta 8, con esto se vuelve a poner de manifiesto que un número elevado de aciertos no implica un resultado mejor, teniéndose como mejores resultados los que reducen en mayor medida el tiempo de búsqueda. Esta independencia del número de aciertos respecto a la bondad del resultado también se hace patente en los mejores resultados ya que el número de aciertos sobre la trayectoria calculada con la heurística para los casos que producen una mejora sobre ésta (zona sombreada en verde) oscila entre 2 y 8. Cabe destacar que se tienen incluso 2 casos dentro del grupo de aceptables que no han tenido ningún acierto respecto a la heurística.

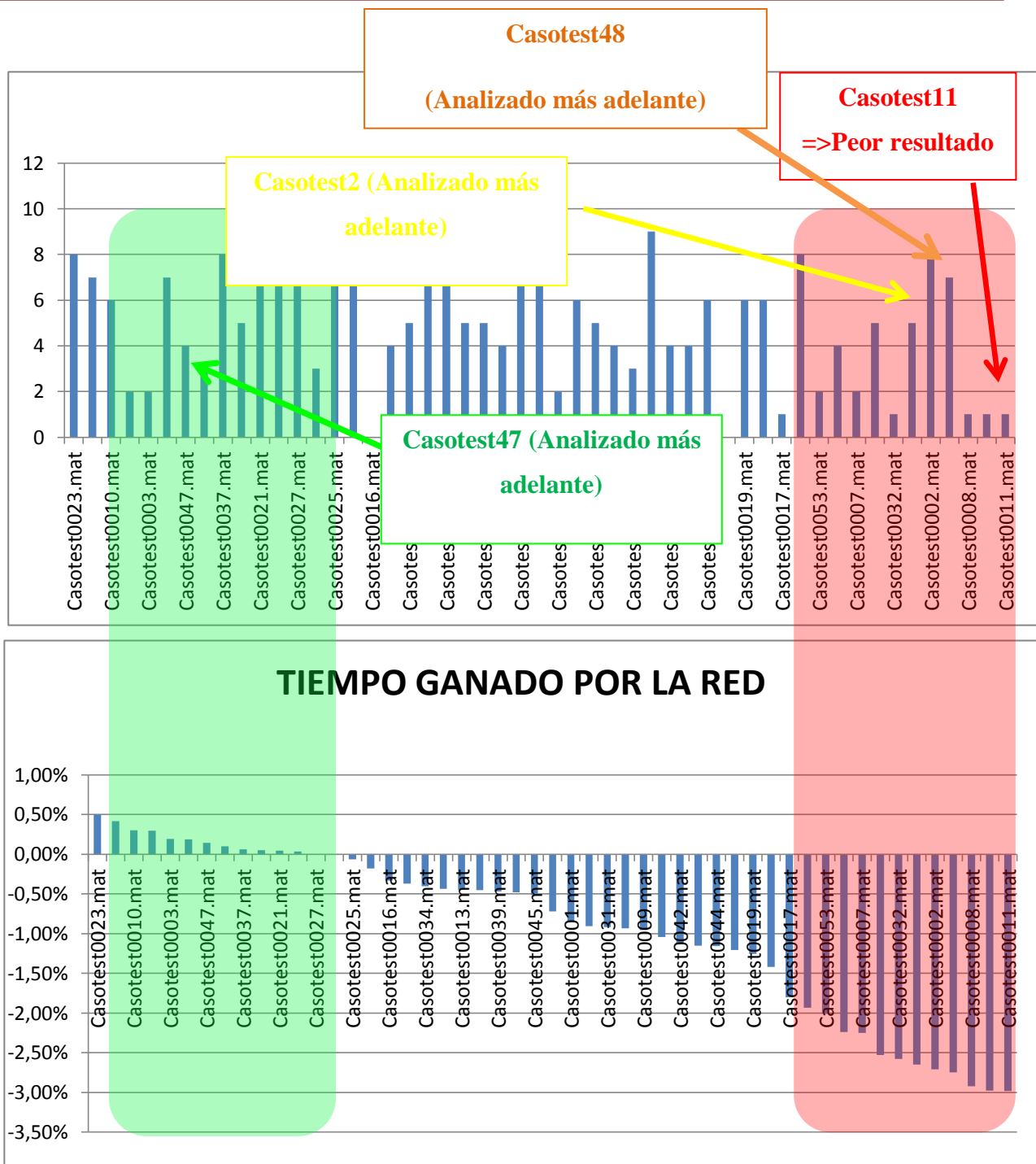
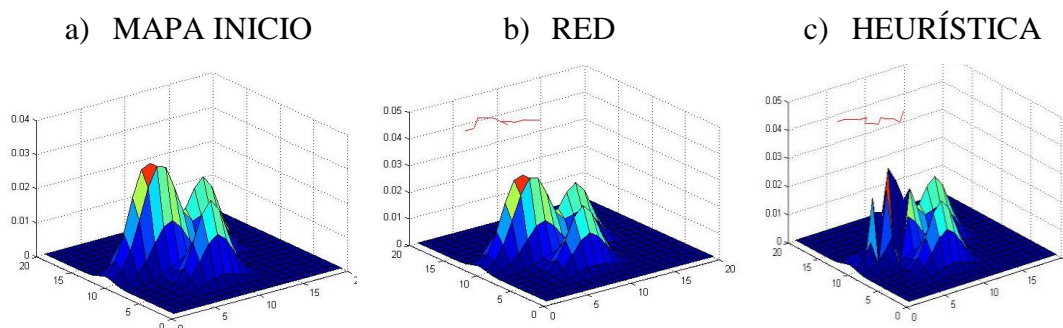


Figura 37: Resultado del procesamiento del conjunto de test "Número de aciertos vs. Tiempo"

La comparativa de trazados entre la red y la heurística para el caso que obtuvo peor resultado "casotest11" (señalizado en la Figura 37) se muestra en la Figura 38, donde se puede ver que la red, a pesar de seguir una dirección parecida a la heurística no consigue acumular probabilidad de detección ya que ésta se

concentra en un pequeño número de celdas. En este ejemplo, la red no ha reproducido el comportamiento de la heurística ya que sólo ha tenido un acierto.



**Figura 38:** Comparativa de los trazados de red y heurística para el ejemplo con peor resultado: Casotest11

Como ya se viene comentando desde el comienzo de este capítulo, el número de aciertos de la predicción de la red respecto al cálculo heurístico no está relacionado directamente con la bondad del resultado. Por lo tanto, a continuación se analizarán dos casos en los que a pesar de tener un número elevado de aciertos, la bondad del resultado es tan baja que están situados entre los casos peores descartados.

El primero de estos casos se muestra en la Figura 39 (ver situación en Figura 37) que se corresponde con el caso de test número 48. En este ejemplo el número de aciertos es 7, pero en la red no se recogen todas las casillas con creencias elevadas, lo que ocasiona una pérdida de tiempo respecto a la heurística del 2,75%.



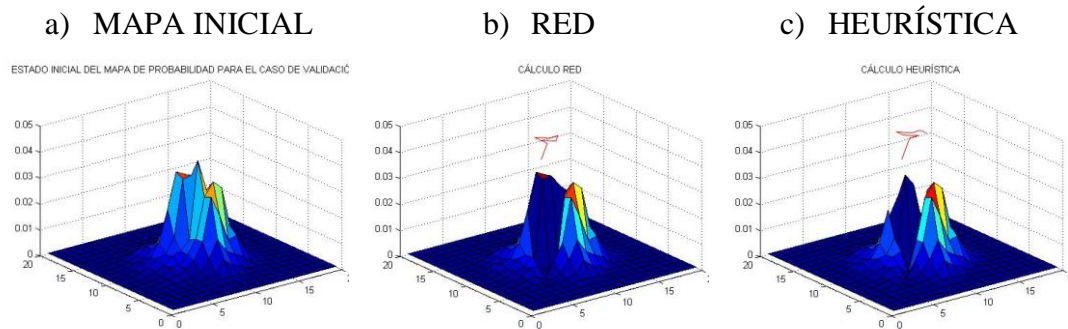


Figura 39: Comparativa de las trayectorias calculadas mediante red y heurística en uno de los peores casos: caso de test 48

El segundo de estos casos “casotest0002” se muestra en la figura 40 (ver situación en Figura 37), a pesar de tener 8 secuencias de control en común con la heurística la medida de LET para la trayectoria calculada por la red es muy inferior a la de la heurística produciéndose en la trayectoria calculada por la red una pérdida de tiempo LET del 2,7%.

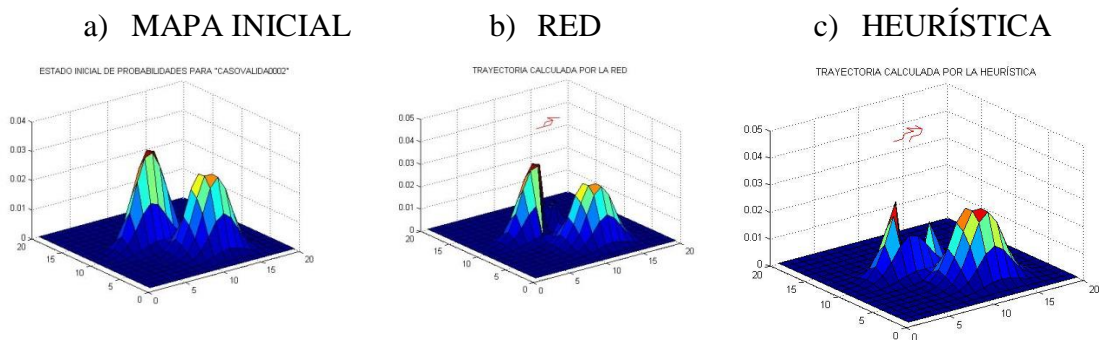


Figura 40: Comparativa de las trayectorias de búsqueda para el cálculo con la red y mediante heurística para el caso de test 2

La independencia del número de aciertos en la trayectoria respecto a la heurística y la bondad del resultado (medida según la Ecuación 1) también se hace patente en los mejores resultados. Una muestra de estos es el caso de test número 47 (ver situación respecto al resto de los casos en la Figura 37) que se muestra en la Figura 41. Este ejemplo tiene una reducción del tiempo mínimo respecto a la heurística de 0,14%, lo que supone una mejora. Sin embargo, el número de aciertos respecto a la heurística es de 4.

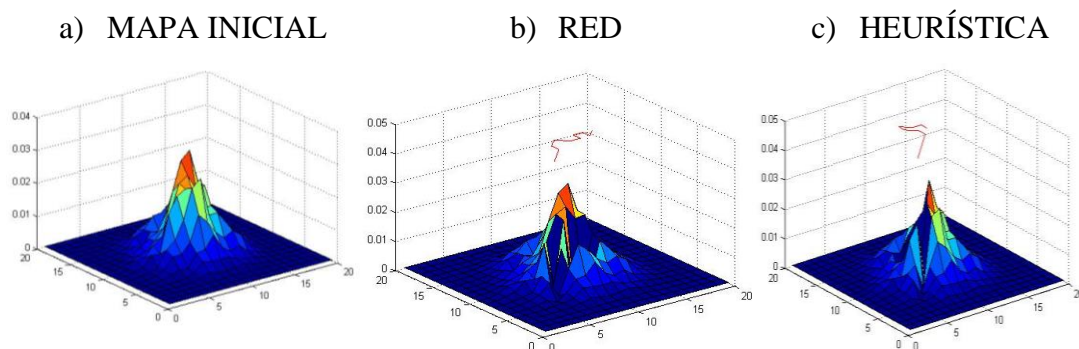


Figura 41: Ejemplo para el que el cálculo efectuado por la red mejora la medida LET: caso de test 47

Las conclusiones de este análisis sobre el conjunto de test se pueden resumir en la existencia de cierta independencia entre la métrica relacionada con el número de aciertos respecto a la heurística y la bondad del resultado (calculada con la ecuación 1). Esto es muy importante ya que demuestra no sólo que la red es capaz de extrapolar trayectorias en ejemplos para los que no ha sido entrenada sino que es capaz, en algunos casos, de mejorar el resultado obtenido con la heurística.

#### 4.3.2 Comparativa de los resultados obtenidos con CEO y con la red neuronal

En este estudio se ha desarrollado una red neuronal que aproxima, mediante una función no lineal, el comportamiento de búsqueda de una heurística calculada con CEO. La red que se ha obtenido cumple los objetivos marcados para ella, pero es necesario comparar los resultados obtenidos por ésta con los del método CEO.

El primer aspecto a comparar es la bondad de los valores obtenidos con ambos métodos. Los datos que se muestran en la Figura 42 corresponden al conjunto de casos de test. En ella se observa que las trayectorias calculadas por la red neuronal son mejores respecto a la métrica definida en la Ecuación 1 en un 24% de las ocasiones. En el 51% de los casos las trayectorias son semejantes o ligeramente peores ocasionando un aumento máximo del tiempo esperado de búsqueda del 1,5%. Los casos calculados por la red que suponen mayores

aumentos en el tiempo de búsqueda son los que se tienen como no aptos y suponen el 25% del total.

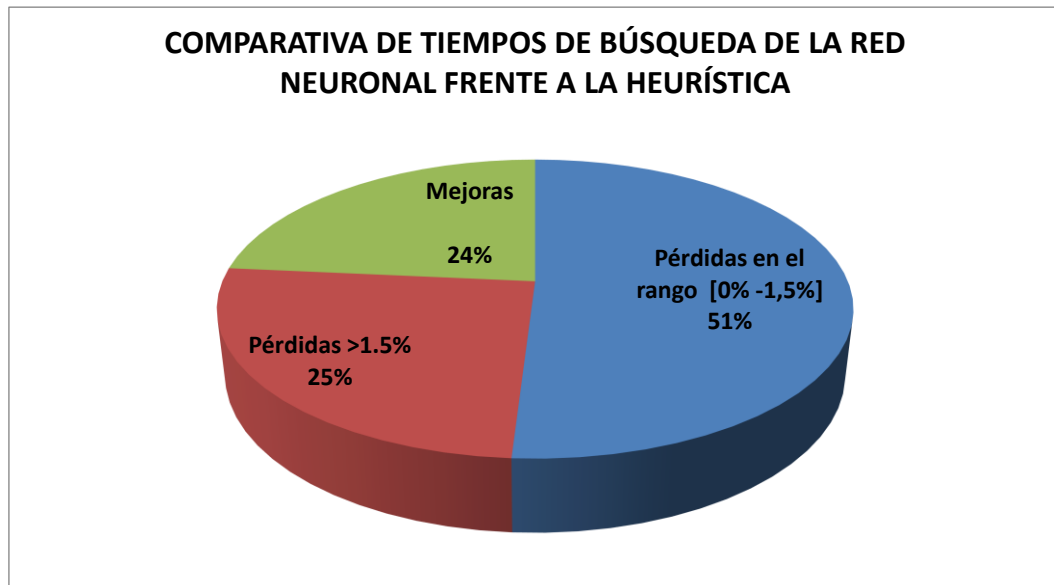


Figura 42: Comparativa del tiempo mínimo calculado mediante red neuronal frente a la heurística

Dado que uno de los requisitos necesarios para el uso del controlador basado en redes neuronales en un cuadrirrotor es su capacidad de respuesta en tiempo real, también se hace necesario analizar el tiempo de respuesta que obtiene la red diseñada y compararlo con el tiempo de respuesta del algoritmo heurístico.

En el algoritmo CEO los tiempos de cálculo de trayectoria oscilan entre cinco a quince segundos. Aunque la red neuronal invierte alrededor de tres horas en su entrenamiento, una vez que se encuentra entrenada, su tiempo de respuesta es inferior a una décima de segundo, tal y como se puede ver en la Figura 43, en la que el 88% de los tiempos de respuesta oscila entre 0,007 y 0,01 segundos. Con estos datos se tiene que el tiempo de respuesta de la heurística puede ser desde 84 hasta 2100 veces superior al necesitado por la red neuronal. Si atendemos a los porcentajes presentados en la Figura 43, en el 88% de los casos el tiempo de

cómputo de la heurística resulta entre 500 y 2100 veces superior al empleado por la red neuronal.

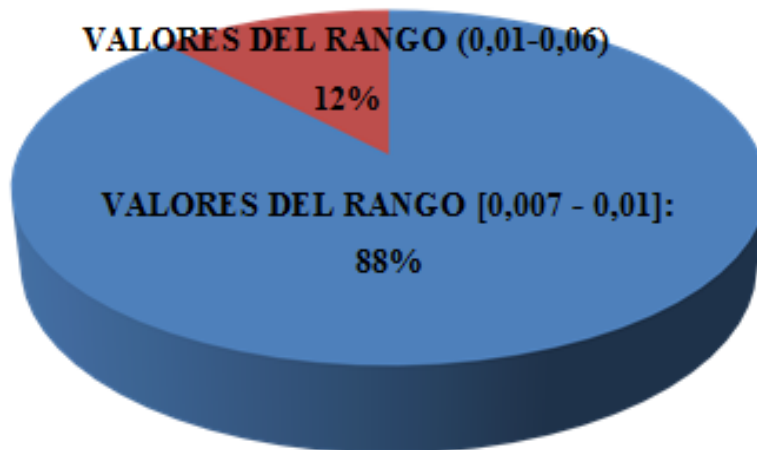


Figura 43. Tiempos invertidos en el cálculo de trayectorias con la red neuronal

Por lo tanto, podemos concluir que mientras que los resultados de la métrica del tiempo de búsqueda obtenidos por la red son similares o no significativamente peores a los obtenidos por la heurística, el tiempo de cómputo de la red neuronal es significativamente inferior al de la heurística.

## 5 Conclusiones

En este capítulo se repasa el cumplimiento de los objetivos establecidos para la red. El estudio se cierra con una sección dedicada a trabajos futuros que se proponen para seguir la investigación en este campo.

### 5.1 Consecución de objetivos

En primer lugar recordemos los objetivos principales definidos a lo largo del estudio.

El objetivo principal es diseñar un controlador basado en redes neuronales que optimice el tiempo de búsqueda de un agente en un entorno probabilístico, capaz de proporcionar respuesta en tiempo real con bajos requerimientos de memoria de forma que éste pueda integrarse en un cuatrirrotor.

Además, la red diseñada debe cumplir los requisitos:

1. **CONJUNTO DE ENTRENAMIENTO.** La red debe obtener resultados aceptables para al menos el 80% de las muestras de este conjunto. Este punto está cumplido ya que los resultados aceptables en el conjunto de entrenamiento después del procesamiento representan el 83%.
2. **CONJUNTO DE VALIDACIÓN.** La cantidad de los resultados aceptables del procesamiento de este conjunto como mucho debe ser un 15% inferior al número de resultados aceptables del conjunto de entrenamiento. Por lo tanto, con los resultados obtenidos para el conjunto de entrenamiento, los resultados del procesamiento del conjunto de validación deben representar un porcentaje de casos descartados inferior al 32%. Este objetivo está cubierto ya que el conjunto de validación tiene un 25% de casos descartados
3. **CONJUNTO DE TEST.** Tiene que obtener unos resultados parecidos a los obtenidos para el conjunto de validación. El objetivo se cumple ya que los

resultados del procesamiento del conjunto de test son los mismos que para el conjunto de validación.

Con esto se tiene que la red neuronal obtenida es capaz de calcular la trayectoria que debe seguir el agente para minimizar el tiempo esperado de búsqueda de un objetivo estático cuya posición está descrita según una distribución de probabilidad.

Las características de la red diseñada son las siguientes:

- Número de entradas: 440 valores de las celdas circundantes al agente
- Número de salidas: 10 secuencias de control
- Tipo de red: Backpropagation
- Número de capas: 3

Tal y como se ha venido exponiendo en puntos anteriores, la bondad de los resultados obtenidos por la red no implica una reproducción exacta de las trayectorias calculadas por la heurística. Esto es muy importante, de hecho, los mejores resultados, que son los que suponen un ahorro en el tiempo de búsqueda de la red respecto a la heurística, son los que tienen un número de movimientos coincidentes menor a ocho.

El hecho de no reproducir exactamente las respuestas calculadas por la heurística hace patente la consecución del objetivo general de cualquier red neuronal: aproximar una función no lineal que cumpla con los casos de entrenamiento. Si la red se limitase a copiar los ejemplos sólo tendríamos una base de datos con una capacidad de extrapolación muy limitada y dado que la casuística de las distribuciones es infinita no tendría utilidad ya que la red nunca podría ser entrenada con todos los ejemplos posibles.

Una muestra de la no reproducibilidad exacta de las secuencias de la heurística se puede ver en la Figura 44. En este caso el número de coincidencias entre la red y

la heurística es de dos, a pesar de esto, la trayectoria de la red produce una mejoría de 0,30% en el tiempo de búsqueda calculado.

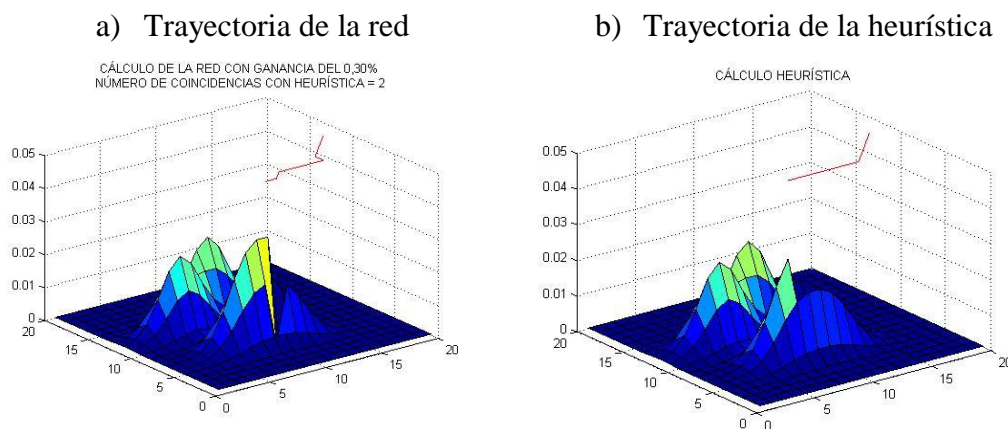


Figura 44: Comparativa de los cálculos de trayectorias mediante red y heurística con mejoría de la red

La principal dificultad encontrada en el entrenamiento de la red ha sido el filtrado de ejemplos para el entrenamiento ya que al generarse éstos aleatoriamente con soluciones no necesariamente óptimas, las trayectorias asociadas pueden variar, confundiendo al proceso de entrenamiento. Se ha identificado que las tipologías más conflictivas son las distribuciones simétricas y las que están alejadas del agente móvil, las cuales se han excluido del entrenamiento según se ha expuesto con anterioridad.

Los casos sobre los que se ha tenido que reforzar el entrenamiento son los que implican varias distribuciones Gaussianas, siendo éstos especialmente delicados por tener las altas probabilidades concentradas en un reducido número de celdas. En estos casos la desviación de una casilla del grid puede implicar la pérdida de una probabilidad importante.

## 5.2 Trabajo futuro

A pesar de la consecución de los objetivos marcados para este estudio, es mucho el trabajo que queda por hacer, sin olvidar que el objetivo final de los trabajos que se han venido desarrollando en esta línea en los últimos años [Fernández Ramírez, 2010; Sáez Martínez, 2012; Lanillos, 2013] es la integración de controladores de trayectoria en cuatrirrotores para que éstos puedan actuar de forma autónoma coordinando las tareas de búsqueda de varios UAVs para encontrar un objetivo móvil en tiempo mínimo.

Para hacer todo esto, van a ser necesarios los avances que se detallan a continuación:

1. Automatización del proceso de generación y filtrado de ejemplos. Tal y como se ha expuesto con anterioridad, las distribuciones simétricas se han filtrado como ejemplos de entrenamiento debido a la dificultad de controlar los ejemplos de entrada con soluciones contradictorias. Lo que les pasa a las distribuciones simétricas puede llegar a pasar con otro tipo de mapas de probabilidad si se multiplican los ejemplos de entrenamiento. Para incluir las distribuciones simétricas en los conjuntos de entrenamiento y aumentar el conjunto de casos de entrenamiento de manera fiable, es necesario filtrar los ejemplos que se generan, de forma que los que impliquen decisiones contradictorias susceptibles de confundir a la red queden automáticamente eliminados. Esto ayudará a ampliar de forma rápida y fiable el elenco de casos contemplados e incluso se mejorarían los resultados obtenidos para los casos contemplados en este estudio ya que como se expuso en el punto 5.1 de este trabajo también pueden existir ejemplos con distribución no simétrica pero parecidos entre sí cuyas trayectorias tengan diferencias significativas.
2. Adaptación al problema dinámico. En este trabajo se aborda el cálculo de trayectorias para el caso estático, esto es, para el caso en el que la posición del objetivo no varía respecto al tiempo o puede considerarse constante



durante todo el horizonte de búsqueda. En el caso general, los objetivos son dinámicos, con lo que generalización para este tipo de problemas se hace imprescindible. Para conseguir esto, se podría concebir el problema dinámico como una sucesión de problemas estáticos sobre los que se calcularían sucesivamente trayectorias de  $N$  movimientos.

3. Coordinación de varios agentes. A lo largo de todo el trabajo se ha considerado un solo agente de búsqueda. Sin embargo, es necesario coordinar varios vehículos no tripulados para que trabajen en la misma zona de forma que sus trayectorias no interfieran entre sí y minimicen de forma conjunta el tiempo de búsqueda.

## BIBLIOGRAFÍA

---

[Berrú Correa, 2013] Berrú Correa, G.B. (2013). Sistemas UAV. <http://www.scribd.com/doc/149129300/Sistemas-Uav> [Accedida en Agosto, 2013]

[Bourgault et al, 2003] Bourgault, F., Furukawa, T., y Durrant-Whyte, H. F. (2003). Optimal search for a lost target in a bayesian world. In FSR, páginas 209-222.

[Bourgault, et al 2004] Bourgault, F., Furukawa, T., y Whyte, H. F. D. (2004). Decentralized Bayesian negotiation for cooperative search. In IEEE Conf. Intelligent robots and systems.

[Eagle, 1984] Eagle, J. N. (1984). The optimal search for a moving target when the search path is constrained. *Operations Research*, 32(5):1107-1115.

[Eagle et al, 1990] Eagle, J. N. y Yee, J. R. (1990). An optimal branch-and-bound procedure for the constrained path, moving target search problem. *Oper. Res.*, 38:110-114.

[Fernández Ramírez, 2010] Fernández Ramírez, F. (2010). Sistemas para la Predicción de Posición y Seguimiento de un Conjunto de Náufragos basado en Redes Neuronales. Proyecto Fin de Máster dirigido por: López Orozco, J.A.

[Fernández Ramírez et al, 2011] Fernández Ramírez, F., Sánchez Benitez, D., Besada Portas, E. y López Orozco, J.A. (2011). Coordinated sea rescue system based on unmanned air vehicles and surface vessels. IEEE Oceans. Junio 2011

[Gan et al, 2010] Gan, S. K. y Sukkarieh, S. (2010). Multi-UAV target search using explicit decentralized gradient-based negotiation. In Proc. IEEE International Conference on Robotics and Automation.

[Hsu et al., 2008] Hsu, D., Lee, W. S., y Rong, N. (2008). A point-based POMDP planner for target tracking. In Proc. IEEE Int. Conf. Robotics and Automation ICRA 2008, páginas 2644-2650.

[Kaelbling et al., 1998] Kaelbling, L. P., Littman, M. L. y Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. Artif. Intell., 101:99-134.

[Lanillos, 2013] Lanillos Pradas, P. (2013). Búsqueda de objetivos móviles en tiempo mínimo en entornos con incertidumbre. Tesis doctoral. Departamento de Arquitectura de Computadores y Automática, Facultad de Informática, Universidad Complutense de Madrid.

[Mathews et al, 2007] Mathews, G., Durrant-Whyte, H. y Prokopenko, M. (2007). Asynchronous gradientbased optimisation for team decision making. In Decision and Control, 2007 46<sup>th</sup> IEEE Conference, 3145-3150.

[Pajares et al, 2007] Pajares, G. y de la Cruz, J.M. (2007). Visión por computador: imágenes digitales y aplicaciones; 2<sup>a</sup> edición, RA-MA, Madrid. 2007.

[Sáez Martínez, 2012] Sáez Martínez, E. (2012). Sistema de rescate de naufragos para la coordinación de vehículos aéreos. Proyecto Fin de Máster dirigido por Besada Portas, E. y López Orozco J.A.

[Sarmiento et al, 2009] Sarmiento, A., Murrieta-Cid, R., y Hutchinson, S. (2009). An efficient motion strategy to compute expected-time locally optimal continuous search paths in known environments. *Advanced Robotics*, 23(12-13):1533-1560.

[Stone, 1975] Stone, L. D. (1975). *Theory of optimal search*. Academic Press, New York.

[Stromquist, 2005] Stromquist, W. (2005). Search for Lost Objects. At Bryn Mawr College, 2005.

[Trummel et al, 1986] Trummel, K. E. y Weisinger, J. R. (1986). The complexity of the optimal searcher path problem. *Operations Research*, 34(2):324-327.

[Wagner, 2010] WAGNER TECHNOLOGIES (2010)  
<http://www.wagner.com/technologies/missionplanning/search-optimization/famous.html> [Accedida en Agosto, 2013]

[Washburn, 1998] Washburn, A. R. (1998). Branch and bound methods for a search problem. *Naval Research Logistics (NRL)*, 45(3):243-257.