

6. OPTIMIZACIÓN DE FÁRMACOS

6.1. INTRODUCCIÓN

En el diseño de un fármaco se busca unas variables de entrada que optimice unas respuestas o características de dicho fármaco. Un método muy empleado por los farmacéuticos para realizar esta optimización es realizar una serie de experimentos utilizando el método DOE y con los datos obtenidos realizar una regresión lineal para obtener los resultados de la optimización.

Como alternativa a la regresión lineal, se estudiará la posibilidad de optimizar fármacos empleando redes neuronales donde cada experimento realizado en el laboratorio será un patrón de entrenamiento o validación de la red. Las entradas a la red neuronal serán las variables de entrada y las respuestas serán las salidas.

En este apartado se llevarán a cabo distintos experimentos para comprobar la viabilidad de emplear redes neuronales para resolver el problema de optimización de fármacos.

6.2. DATOS

En este proyecto se han utilizado datos experimentales del diseño de dos fármacos. Los datos del primer fármaco, Tabla 6.1, proceden de una batería de problemas del Laboratorio de Técnicas Farmacéuticas de la Facultad de Farmacia de Cluj-Napoca, Rumanía [1], [2], [3]. Estos datos consta de once experimentos con cinco entradas: la velocidad de la turbina de mezclado, la concentración del aglutinante, la velocidad de adición, la proporción de talco y la proporción de lauril sulfato sódico. Estas variables se denominarán X_1 , X_2 , X_3 , X_4 y X_5 . Para cada fármaco se midieron seis

parámetros o respuestas: la función de carga, diámetro medio del comprimido, el valor Carr, el valor Haussner, el tiempo de flujo y la fragilidad. Estas respuestas se denotarán por Y_1 , Y_2 , Y_3 , Y_4 , Y_5 , e Y_6 .

Exp. Nº	Variables de Entrada					Respuestas (Salidas)					
	X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6
1	20	2	3	5	1	84.0	973.8	42.2	1.043	7.85	1.165
2	40	2	3	0	0	71.9	1150.0	1.6	1.016	8.2	2.264
3	20	8	3	0	1	92.5	1121.4	4.2	1.044	8.83	0.700
4	40	8	3	5	0	88.1	1200.0	3.7	1.038	8.87	1.205
5	20	2	9	5	0	99.2	910.0	5.8	1.061	8.3	1.914
6	40	2	9	0	1	68.2	985.1	4.1	1.043	7.9	2.550
7	20	8	9	0	0	99.1	1010.0	5.3	1.056	9.05	1.160
8	40	8	9	5	1	83.9	925.4	5.5	1.058	8.5	1.265
9	30	5	6	2.5	0.5	85.0	1055.8	3.8	1.036	8.3	1.535
10	30	5	6	2.5	0.5	81.2	1030.0	4.1	1.042	8.37	1.490
11	30	5	6	2.5	0.5	85.0	1060.0	4.1	1.042	8.4	1.535

Tabla 6.1: Datos experimentales de la Facultad de Farmacia de Cluj-Napoca.

Los datos de segundo fármaco se han obtenido de un experimento real procedente del Dpto. de Farmacia y Tecnología Farmacéutica de la Universidad de Sevilla. Se dispone de datos de 18 experimentos diferentes y cada experimento está formado por 8 variables de entrada y 16 respuestas. Las variables de entrada son en este caso: cantidad de TEA (gotas), concentración de carbopol (%), velocidad de flujo de caída (ml/min), velocidad de agitación (r.p.m.), tiempo de reacción con los iones Ca^{2+} (min), tipo de secado (ambiente, rotavapor y estufa), adición de electrolito (NaCl M) y el pH del medio. Las variables de salida son: T50% (tiempo que se tarda en liberar el 50% de la dosis), T420% (porcentaje de fármaco liberado a los 420 minutos), EE (eficacia de encapsulación), hinchamiento, desorción, AUC 360 (área bajo la curva de liberación), THTOmax (tiempo que se tarda en alcanzar el hinchamiento máximo), relación de aspecto, circularidad, dimensión fractal, heterogeneidad, factor de forma, área, diámetro medio, perímetro, D Eq. Fere (Diámetro equivalente de Feret).

En la Tabla 6.2 se muestran las variables de entrada de los 18 experimentos y en la Tabla 6.3 y en la Tabla 6.4 están las respuestas del 1-8 y del 9-16 respectivamente.

Exp Nº	Variables de Entrada							
	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈
1	0	1	1	100	5	Ambiente	0	7.2
2	0	1	2	300	30	Rotavapor	0.01	7.6
3	0	1	3	500	60	Estufa	0.1	8
4	0	2	1	100	30	Rotavapor	0.1	8
5	0	2	2	300	60	Estufa	0	7.2
6	0	2	3	500	5	Ambiente	0.01	7.6
7	0	4	1	300	5	Estufa	0.01	8
8	0	4	2	500	30	Ambiente	0.1	7.2
9	0	4	3	100	60	Rotavapor	0	7.6
10	4	1	1	500	60	Rotavapor	0.01	7.2
11	4	1	2	100	5	Estufa	0.1	7.6
12	4	1	3	300	30	Ambiente	0	8
13	4	2	1	300	60	Ambiente	0.1	7.6
14	4	2	2	500	5	Rotavapor	0	8
15	4	2	3	100	30	Estufa	0.01	7.2
16	4	4	1	500	30	Estufa	0	7.6
17	4	4	2	100	60	Ambiente	0.01	8
18	4	4	3	300	5	Rotavapor	0.1	7.2

Tabla 6.2: Variables de entrada de los datos experimentales de la Facultad de Farmacia de Sevilla.

Las variables en texto se codifican en números, como por ejemplo:

- Ambiente: 1
- Rotavapor: 2
- Estufa: 3

Exp Nº	Respuestas I (Salidas)							
	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	Y ₈
1	73,00	95,99	72,64	2863,78	23,13	26981,80	330,00	1,61
2	50,00	96,38	82,32	1937,37	30,03	28510,41	240,00	1,56
3	69,33	81,57	80,44	1320,67	28,72	24289,11	180,00	1,71
4	91,33	75,43	82,31	1681,62	22,85	21875,67	330,00	1,55
5	102,33	93,83	88,50	2177,52	24,65	24036,90	300,00	1,72
6	198,33	84,50	80,27	1119,52	20,67	16481,53	180,00	1,34
7	118,67	92,30	81,51	1610,31	21,18	21745,19	300,00	2,02
8	293,33	81,50	81,70	1082,83	16,80	14506,47	210,00	1,49
9	106,00	71,50	85,71	1459,06	18,90	20058,25	180,00	1,57
10	60,00	85,64	89,59	2786,18	23,09	25366,97	360,00	1,29
11	82,67	85,01	88,99	1151,14	19,69	24478,79	240,00	1,30
12	119,33	75,16	84,11	3685,23	19,34	20363,20	300,00	1,14
13	111,67	76,00	88,25	1868,79	19,89	20122,69	360,00	1,69
14	148,67	76,22	89,82	3003,21	18,37	18598,00	300,00	2,51
15	209,33	65,02	89,23	2017,31	21,47	16740,94	180,00	1,22
16	238,67	58,54	80,96	3094,41	17,10	14550,95	300,00	1,88
17	142,00	91,59	84,58	1554,57	16,79	22033,27	300,00	1,40
18	138,33	67,50	90,03	1420,16	20,66	19199,98	180,00	2,21

Tabla 6.3: 1-8 variables de salida de los datos experimentales de la Facultad de Farmacia de Sevilla.

Exp Nº	Respuestas II (Salidas)							
	Y ₉	Y ₁₀	Y ₁₁	Y ₁₂	Y ₁₃	Y ₁₄	Y ₁₅	Y ₁₆
1	2,73	1,15	0,45	0,52	1,43	1,34	5,60	1,56
2	3,36	1,16	0,45	0,20	1,57	1,47	9,93	1,98
3	4,88	1,19	0,51	0,25	1,66	1,53	8,94	1,92
4	1,80	1,13	0,47	0,58	2,19	1,72	7,15	1,78
5	3,60	1,20	0,73	0,27	1,91	1,64	9,37	2,03
6	1,34	1,20	0,34	0,65	2,27	1,69	6,05	1,82
7	3,47	1,20	0,47	0,30	1,48	1,51	8,54	1,99
8	6,77	1,07	0,57	0,25	1,67	1,54	9,36	1,91
9	2,15	1,18	0,13	0,55	2,09	1,66	6,69	1,86
10	6,10	1,22	0,22	0,19	2,65	2,07	14,38	2,82
11	3,46	1,26	0,66	0,26	2,28	1,83	9,52	2,20
12	1,71	1,06	0,35	0,55	2,09	1,57	6,25	1,67
13	4,87	1,23	0,46	0,40	2,40	1,73	7,42	1,79
14	5,40	1,14	0,41	0,20	1,65	1,72	10,12	2,44
15	3,26	1,16	0,44	0,38	2,33	1,56	8,17	1,77
16	2,92	1,20	0,53	0,35	2,08	1,67	9,53	1,83
17	3,19	1,17	0,49	0,34	1,56	1,41	6,81	1,57
18	3,05	1,13	0,41	0,26	4,50	2,81	14,46	3,61

Tabla 6.4: 9-16 variables de salida de los datos experimentales de la Facultad de Farmacia de Sevilla.

6.3. PROBLEMAS POR LA INSUFICIENCIA DE DATOS

Para entrenar y evaluar correctamente una red neuronal se recomienda dividir los datos disponibles en dos grupos independientes. El primer grupo de datos se emplea para entrenar la red neuronal y el segundo para validarla. Normalmente el 60-70% de los datos serán para el entrenamiento y el resto para la validación. Es necesario que ambos grupos sean una muestra representativa de la población total, así la red neuronal

aprenderá toda la información necesaria para llevar a cabo su tarea una vez terminado el aprendizaje y dicha red será evaluado con todos los casos posibles que se pueda encontrar.

Cuando el número de datos es insuficiente la muestra no será representativa de la población y nos encontraremos con dificultades tanto en el entrenamiento como en la validación:

- Entrenamiento: Cuando se tiene insuficientes datos, no tenemos una muestra representativa de la población y no se le ofrece a la red neuronal datos suficientes para que aprenda a realizar la tarea deseada correctamente. En este caso la red neuronal aprenderá los casos mostrados durante el entrenamiento ofreciendo un error muy pequeño para éstos y, sin embargo, obtendremos errores grandes ante datos nuevos ya que la red neuronal no es capaz de generalizar.
- Validación: La validación de la red neuronal también requiere una muestra representativa de la población. Si el número de datos es insuficiente puede que sólo se disponga de dos o tres patrones para verificar el error cometido por la red. El error cometido por un patrón o dos no necesariamente es el error real que cometerá la red una vez que se comience a dar uso.

Ante estos problemas se estudió otra forma alternativa de entrenar y validar redes neuronales: el método de remuestreo con reposición *Bootstrap*. Si se utiliza este método para la validación de la red neuronal, se puede entrenar dicha red con todos los datos disponibles y se validará con los mismos datos sin necesidad de dividir los escasos patrones disponibles en dos.

6.4. EVALUACIÓN DEL USO DE LAS REDES NEURONALES

Se han diseñado unos experimentos con el objetivo de comprobar la utilidad de las redes neuronales en optimización de fármacos estimando el error que cometería la red neuronal. Se han comparado dos métodos de validación de redes neuronales entre sí: el método de división de datos para el entrenamiento y la validación con el método de remuestreo *Bootstrap*. Estos experimentos se han repetido dos veces: una vez con los datos publicados en [1], [2], [3]. y otra con los experimentos de la Facultad de Farmacia de la Universidad de Sevilla.

La Facultad de Farmacia de la Universidad de Sevilla, además de los experimentos obtenidos en el laboratorio mediante el Diseño de Experimentos (DOE), nos facilitó los resultados obtenidos del análisis de optimización de dicho fármaco mediante la técnica de regresión lineal. Se comprobará la efectividad de las redes neuronales en la optimización del fármaco comparando los resultados obtenidos mediante el uso de las redes neuronales con los facilitados por el la Facultad de Farmacia de Sevilla.

Antes de continuar con los experimentos es necesario determinar la arquitectura de la red neuronal que se empleará y el algoritmo de aprendizaje con los correspondientes parámetros.

6.4.1. Determinación de la arquitectura y el aprendizaje

La arquitectura de red neuronal que se empleará para optimizar fármacos será la del perceptrón multicapa con tres capas, dado que esta arquitectura es capaz de resolver la mayoría de los problemas prácticos. Como se ha visto en el Apartado 3.2, un perceptrón de dos capas no posee las suficientes conexiones para aprender ciertas

funciones y un perceptrón con dos capas ocultas en la práctica rara vez es necesario, además de que ralentiza el aprendizaje por dos motivos:

- El gradiente del algoritmo de aprendizaje se vuelve más inestable cuanto mayor sea el número de capas.
- El número de mínimos locales aumenta con el número de capas.

Para completar el diseño de la arquitectura del perceptrón multicapa habrá que determinar el número de neuronas en cada capa. El número de neuronas en las capas de entrada y salida vendrán determinado por el problema que queramos resolver: el número de neuronas en la capa de entrada será el número de variables de entrada de los experimentos que se presentarán a la red y la capa de salida tendrá el mismo número de neuronas que el número de respuestas. El número de neuronas de la capa oculta se determinará de manera experimental dado que no existe un método que halle el número óptimo de neuronas en esta capa.

La red neuronal será entrenada mediante el algoritmo de aprendizaje *Backpropagation* con momento. En este algoritmo hay que definir dos parámetros:

- Tasa de aprendizaje: es el tamaño de paso en el algoritmo del gradiente descendiente. Si esta tasa es demasiado pequeña, la velocidad de convergencia del algoritmo será demasiado lento. Por el contrario, si es demasiado grande puede que el error oscile en torno al mínimo y no converja.
- Momento: se añade al algoritmo de aprendizaje un término de momento para suavizar las oscilaciones del error y aumentar la velocidad de convergencia al seguir mejor la dirección del mínimo. Las oscilaciones no se pueden eliminar totalmente al menos que este término sea muy grande. Sin embargo, si el momento es demasiado grande, el algoritmo no podrá seguir adecuadamente la superficie de la función de error.

Para todos los experimentos se tomó el valor de 0.1 como tasa de aprendizaje y un momento de 0.9.

La condición de parada para el aprendizaje será cuando la red neuronal cometa un error menor que 0.01 en los patrones de entrenamiento. Existen otras dos condiciones por las que se detendrá el aprendizaje. El primero de estas condiciones es si el gradiente cae por debajo de un cierto valor. Esto ocurre cuando se alcanza una “planicie” y la pendiente de la función es muy pequeña. En este caso, la búsqueda del mínimo queda estancando y se opta por detener el aprendizaje. La segunda condición es cuando el entrenamiento supera un número de iteraciones. Este parámetro es para poner un límite máximo en la duración del entrenamiento. Los valores de estos dos parámetros en este proyecto se han establecido en 0.00001 para el gradiente mínimo y 10.000 como número de iteraciones máximas.

6.4.1.1. Determinación del número de neuronas ocultas

Queda el establecer el número de neuronas de la capa oculta de la red neuronal para terminar de definir la arquitectura. El número de neuronas en la capa oculta influye mucho en el comportamiento de la red neuronal tanto durante el entrenamiento como después cuando se ponga en funcionamiento. Si el número de neuronas es insuficiente, la red neuronal no tendrá la suficiente capacidad para aprender a resolver el problema. Si, por el contrario, la capa oculta tiene demasiadas neuronas el aprendizaje se puede alargar excesivamente.

Otro problema que surge por un número excesivo de neuronas en la capa oculta es el sobreentrenamiento. Una red neuronal con muchas neuronas posee una gran capacidad de aprendizaje por lo que comenzará a considerar información superflua para la aplicación. Esto dará como resultado una red neuronal que ofrece un error pequeño para los patrones de entrenamiento pero que da errores grandes para patrones nuevos, esto es, ha perdido la capacidad de generalizar.

La mejor forma para determinar el número de neuronas óptima para la capa oculta es comenzar con un número pequeño de neuronas, por ejemplo dos, y entrenar y

validar la red. Si el resultado no es satisfactorio se eleva el número de neuronas y se vuelve a entrenar y validar. Este proceso se repite hasta que el error cometido por la red sea pequeño o hasta que el aumentar el número de neuronas no mejore significativamente el resultado de la red.

El número óptimo de neuronas será distinto para cada aplicación. Por ello, este proceso se repetirá para las dos muestras de datos experimentales de las que disponemos: los datos de la Facultad de Farmacia de Cluj-Napoca [1], [2], [3]. (Datos I), y los datos facilitados por la Facultad de Farmacia de la Universidad de Sevilla (Datos II).

Datos I:

Primero se determinará el número de neuronas de la capa oculta óptima para resolver el problema de optimización del fármaco cuyos experimentos provienen de la Facultad de Farmacia de Cluj-Napoca [1], [2], [3]. Estos datos están formados por 11 experimentos con 5 variables de entrada y 6 respuestas.

Todas las redes neuronales son perceptrones multicapa con 5 neuronas de entrada y 6 neuronas de salida. El número de neuronas de la capa oculta se variará de 1 a 15. Para cada arquitectura se entrena la red neuronal con los patrones de entrenamiento y se comprueba el error cometido por la red neuronal en los patrones de validación.

Los pesos de las redes neuronales se han inicializado de manera aleatoria con la instrucción *random* del lenguaje de programación C. Para poder realizar una mejor comparativa, se han utilizado 5 semillas distintas para esta instrucción. Todas las redes se han entrenado con el algoritmo de *Backpropagation* con una tasa de aprendizaje de 0.1 y un momento de valor 0.9. El aprendizaje se ha detenido cuando el error en los patrones de entrenamiento ha alcanzado el valor de 0.01 o cuando el algoritmo ha realizado 10.000 iteraciones.

Los resultados se muestran de la Tabla 6.5 a la Tabla 6.7. Todas las redes son evaluadas mediante el error RMS de los patrones de entrenamiento y de validación. En

la primera tabla se ha entrenado con el 70% de los 11 datos experimentales disponibles y se ha validado con el 30% restante de dichos datos. En la segunda tabla se muestra los resultados de haber entrenado con el 80% de los datos y validado con el 20%. En la tercera tabla se entrenó las redes con el 90% de los datos y validado con el 10%.

	N° NEURONAS														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RMS entr	0.0450	0.0207	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099	0.0099
RMS valid	0.123886	0.0089199	0.083125	0.0096285	0.0094352	0.088945	0.089368	0.102027	0.0096233	0.0097162	0.0101271	0.0095455	0.1001770	0.1001691	0.0108945
RMS entr	0.04504	0.02073	0.001000	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999
RMS valid	0.117848	0.0094189	0.0089389	0.080115	0.0096452	0.0092614	0.0092634	0.078105	0.0094401	0.0091561	0.0088787	0.0099167	0.0092836	0.0091237	0.009971
RMS entr	0.0450	0.0207	0.0099	0.001000	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999
RMS valid	0.117702	0.0093206	0.090149	0.000173	0.0094921	0.0094582	0.089880	0.0095183	0.0094780	0.0094529	0.00945181	0.0094121	0.0093718	0.0098812	0.010094
RMS entr	0.0450	0.0207	0.0099	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999
RMS valid	0.117290	0.010086	0.085158	0.00927	0.0096553	0.0095545	0.0095885	0.0092623	0.0094247	0.0094007	0.00945648	0.091471	0.0096210	0.0096092	0.0097374
RMS entr	0.0450	0.0207	0.0099	0.000999	0.000999	0.000999	0.000999	0.000999	0.001000	0.000999	0.000999	0.000999	0.000999	0.000999	0.000999
RMS valid	0.118049	0.0100901	0.080931	0.0091448	0.0006270	0.0092081	0.0088432	0.0094108	0.0094293	0.0094912	0.00985470	0.0089037	0.0094413	0.0094005	0.086461
RMS Media	0.1189	0.00955	0.08575	0.009237	0.009771	0.009275	0.09124	0.009241	0.009479	0.009523	0.009950	0.009385	0.009579	0.009637	0.00978

Tabla 6.5: Errores en entrenamiento y validación con entrenando con el 70% de los datos.

6. OPTIMIZACIÓN DE FÁRMACOS

	N° NEURONAS														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RMS entr	0.0 44 97	0.0 23 95	0.0 11 03	0.0 09 99	0.0 99 98	0.0 09 99	0. 00 99	0.0 09 99	0.0 09 99	0.0 09 99	0.0 10 00	0. 00 99	0. 00 99	0.0 10 00	0. 00 99
RMS valid	0.1 11 33 7	0.0 37 22 0	0.0 47 03 5	0.0 44 08 3	0.0 44 25 0	0.0 63 96 7	0. 04 42 75	0.0 45 83 6	0.0 41 53 0	0.0 49 08 5	0.0 48 52 4	0. 04 61 89	0. 05 45 21	0.0 48 89 4	0. 04 69 08
RMS entr	0.0 44 97	0.0 23 95	0.0 11 03	0.0 10 00	0.0 09 99	0.0 09 99	0. 00 99	0.0 09 99	0.0 09 99	0.0 09 99	0.0 09 99	0. 01 00	0. 00 99	0.0 09 99	0. 00 99
RMS valid	0.1 11 32 0	0.0 37 10 9	0.0 65 44 5	0.0 56 25 2	0.0 44 89 9	0.0 41 31 2	0. 04 98 98	0.0 40 05 6	0.0 50 62 0	0.0 49 86 3	0.0 43 77 5	0. 05 93 61	0. 04 82 52	0.0 52 67 0	0. 05 15 06
RMS entr	0.0 44 97	0.0 23 95	0.0 11 03	0.0 09 99	0.0 09 99	0.0 09 99	0. 00 99	0.0 10 99	0.0 09 99	0.0 10 00	0.0 09 99	0. 00 99	0. 01 00	0.0 09 99	0. 00 99
RMS valid	0.1 11 31 7	0.0 36 02 6	0.0 55 54 3	0.0 42 11 9	0.0 49 80 3	0.0 51 29 5	0. 04 10 41	0.0 54 55 5	0.0 49 91 1	0.0 42 07 1	0.0 51 60 6	0. 04 84 64	0. 04 72 14	0.0 47 09 6	0. 05 13 97
RMS entr	0.0 44 97	0.0 23 95	0.0 11 03	0.0 10 00	0.0 09 99	0.0 09 99	0. 00 99	0.0 10 99	0.0 09 99	0.0 10 00	0.0 09 99	0. 00 99	0. 01 00	0.0 09 99	0. 00 99
RMS valid	0.1 11 33 3	0.0 36 54 4	0.0 45 26 2	0.0 48 26 2	0.0 49 60 5	0.0 48 20 9	0. 04 81 35	0.0 61 40 2	0.0 48 19 2	0.0 51 80 2	0.0 44 27 1	0. 04 88 58	0. 05 85 63	0.0 49 31 1	0. 06 04 61
RMS entr	0.0 44 97	0.0 23 95	0.0 11 03	0.0 09 99	0.0 10 00	0.0 09 99	0. 01 00	0.0 09 99	0.0 09 99	0.0 09 99	0.0 09 99	0. 00 99	0. 00 99	0.0 09 99	0. 01 00
RMS valid	0.1 11 32 5	0.0 37 49 8	0.0 53 04 9	0.0 46 69 8	0.0 43 59 9	0.0 58 32 8	0. 04 50 76	0.0 45 99 8	0.0 47 39 3	0.0 49 24 5	0.0 49 31 1	0. 04 72 83	0. 05 19 90	0.0 49 23 2	0. 05 11 58
RMS Media	0.1 11 33	0.0 36 88	0.0 53 27	0.0 47 60	0.0 46 43	0.0 52 62	0. 04 56	0.0 49 57	0.0 47 53	0.0 48 41	0.0 47 50	0. 05 00	0. 05 21	0.0 49 44	0. 05 22

Tabla 6.6: Errores en entrenamiento y validación con entrenando con el 80% de los datos.

6. OPTIMIZACIÓN DE FÁRMACOS

	N° NEURONAS														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RMS entr	0.0 52 03	0.0 24 46	0.0 13 22	0.0 10 00	0.0 09 99	0.0 09 99	0.0 09 99	0.0 09 99	0. 00 99	0.0 09 99	0.0 10 00	0.0 09 99	0.0 09 99	0.0 10 00	0.0 09 99
RMS valid	0.0 76 55 9	0.0 21 14 7	0.0 55 32 2	0.0 51 45 8	0.0 26 45 8	0.0 40 83 6	0.0 34 55 7	0.0 39 89 2	0. 04 72 09	0.0 48 81 0	0.0 43 41 8	0.0 47 31 3	0.0 47 77 9	0.0 38 44 6	0.0 56 85 4
RMS entr	0.0 52 03	0.0 24 46	0.0 13 22	0.0 10 00	0.0 09 99	0.0 10 00	0.0 10 00	0.0 09 99	0. 01 00 0	0.0 09 99	0.0 09 99	0.0 10 00	0.0 09 99	0.0 09 99	0.0 10 00
RMS valid	0.0 76 60 1	0.0 20 18 3	0.0 58 83 4	0.0 28 63 0	0.0 53 88 1	0.0 37 95 9	0.0 36 55 5	0.0 36 11 0	0. 04 02 79	0.0 46 98 0	0.0 44 34 6	0.0 42 69 4	0.0 52 80 8	0.0 49 05 8	0.0 44 79 1
RMS entr	0.0 52 03	0.0 24 46	0.0 13 22	0.0 09 99	0.0 10 00	0.0 09 99	0.0 09 99	0.0 09 99	0. 00 99	0.0 09 99	0.0 09 99	0.0 09 99	0.0 10 00	0.0 10 00	0.0 09 99
RMS valid	0.0 76 63 0	0.0 25 69 7	0.0 28 31 4	0.0 27 90 1	0.0 44 65 8	0.0 36 49 2	0.0 34 59 3	0.0 47 16 7	0. 04 29 92	0.0 38 73 8	0.0 47 15 9	0.0 44 93 9	0.0 44 70 8	0.0 40 38 7	0.0 50 10 8
RMS entr	0.0 52 03	0.0 24 46	0.0 13 22	0.0 09 99	0.0 10 00	0.0 10 00	0.0 09 99	0.0 09 99	0. 00 99	0.0 09 99	0.0 09 99	0.0 09 99	0.0 09 99	0.0 09 99	0.0 09 99
RMS valid	0.0 76 43 2	0.0 08 00 9	0.0 37 69 2	0.0 44 71 4	0.0 59 02 4	0.0 44 88 2	0.0 43 42 6	0.0 47 84 8	0. 04 72 81	0.0 43 26 8	0.0 36 93 7	0.0 48 10 5	0.0 48 29 9	0.0 35 25 8	0.0 52 44 4
RMS entr	0.0 52 03	0.0 24 46	0.0 13 22	0.0 10 00	0.0 10 00	0.0 10 00	0.0 09 99	0.0 09 99	0. 00 99	0.0 09 99	0.0 09 99	0.0 09 99	0.0 10 00	0.0 09 99	0.0 10 00
RMS valid	0.0 76 58 6	0.2 26 96 0	0.0 35 96 0	0.0 53 46 3	0.0 54 23 6	0.0 40 05 4	0.0 41 00 5	0.0 40 21 5	0. 04 94 79	0.0 53 42 6	0.0 32 57 1	0.0 51 36 3	0.0 47 27 9	0.0 51 01 2	0.0 52 93 6
RMS Media	0.0 76 56	0.0 60 40	0.0 43 22	0.0 41 23	0.0 47 65	0.0 40 04	0.0 38 03	0.0 42 24	0. 04 54	0.0 46 24	0.0 40 89	0.0 46 88	0.0 48 17	0.0 42 83	0.0 51 43

Tabla 6.7: Errores en entrenamiento y validación con entrenando con el 90% de los datos.

En las tablas anteriores cuando el número de neuronas en la capa oculta es menor que cuatro, el error RMS cometido por las redes neuronales en los patrones de entrenamiento es mayor que 0.01. Esto es, el error durante el entrenamiento nunca cae por debajo del error mínimo deseado y el entrenamiento se detiene en el número máximo de iteraciones permitido. Esto es debido a que la red neuronal no posee las suficientes conexiones para aprender a realizar la tarea con un error menor al indicado. Por tanto, el número de neuronas ocultas para este caso será mayor o igual a cuatro.

En la siguiente tabla se muestra el número de neuronas de los dos casos que dieron mejor resultado de error RMS en los patrones de validación. A partir de los resultados se puede deducir que el número óptimo de neuronas en la capa oculta está entre 3 y 7 neuronas. Para este caso se ha optado por 7 neuronas en la capa oculta ya que da buenos resultados en los tres casos.

% Datos en Entrenamiento	Nº Neuronas (1)	Nº Neuronas (2)
70	3	7
80	7	5
90	7	6

Tabla 6.8: Número de neuronas óptimo.

Con esto, la arquitectura de red neuronal para este caso será el perceptrón de tres capas con 5 neuronas en la capa de entrada, 7 neuronas en la capa oculta y 6 neuronas en la de salida.

Datos II:

Para determinar el número óptimo de neuronas de la capa oculta en el problema de optimizar el medicamento desarrollado por La Facultad de Farmacia de Sevilla se

procede de la misma manera que en el caso anterior. En este caso se dispone de 18 experimentos con 8 variables de entrada y 16 respuestas.

Los resultados se muestran de la Tabla 6.9 a la Tabla 6.11. En la primera tabla se muestra los resultados de entrenar la red neuronal con el 70% de los experimentos y validado con el 30% restante, la segunda tabla se muestra los resultados de haber entrenado con el 80% de los datos y validado con el 20% y la última tabla muestra los resultados obtenidos al entrenar con el 90% y validado con el 10% de los datos.

6. OPTIMIZACIÓN DE FÁRMACOS

	N° NEURONAS														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RMS entr	0.12511	0.10157	0.81900	0.06886	0.04844	0.00422	0.03521	0.02541	0.01511	0.00999	0.00999	0.00099	0.00998	0.00996	0.00998
RMS valid	0.199658	0.205489	0.221584	0.301706	0.254948	0.0246412	0.241023	0.253304	0.208583	0.20457	0.197291	0.0193894	0.195088	0.193264	0.193602
RMS entr	0.12512	0.10157	0.007920	0.07296	0.05201	0.04145	0.03386	0.02491	0.01525	0.00999	0.01000	0.00099	0.00999	0.00998	0.01000
RMS valid	0.199647	0.205890	0.244989	0.195367	0.274902	0.0293353	0.208972	0.220772	0.250030	0.191981	0.19909	0.0174109	0.200781	0.200760	0.190891
RMS entr	0.12511	0.10156	0.007757	0.06490	0.05019	0.0416	0.03406	0.02548	0.01510	0.00999	0.00999	0.00099	0.00999	0.00996	0.00998
RMS valid	0.199654	0.202843	0.267046	0.207473	0.241601	0.0331104	0.273771	0.235052	0.197324	0.190873	0.185614	0.019393	0.179440	0.201703	0.196410
RMS entr	0.12396	0.10145	0.008227	0.06676	0.05169	0.0420	0.03450	0.02665	0.01509	0.00999	0.00999	0.00099	0.00998	0.00996	0.00997
RMS valid	0.196732	0.208635	0.219350	0.330126	0.251395	0.0233880	0.258311	0.232919	0.200811	0.2008815	0.190155	0.0204949	0.177588	0.201965	0.19486
RMS entr	0.12492	0.10157	0.007980	0.06376	0.05098	0.0415	0.03361	0.02738	0.01524	0.00900	0.00999	0.00099	0.00998	0.00994	0.00996
RMS valid	0.199670	0.205917	0.256008	0.296562	0.284843	0.02521	0.28181	0.24194	0.20470	0.19376	0.18952	0.02085	0.183902	0.201846	0.196420
RMS Media	0.19907	0.20935	0.24180	0.26625	0.25234	0.02774	0.24618	0.23680	0.22466	0.19640	0.19232	0.01943	0.18736	0.19771	0.19356

Tabla 6.9: Errores en entrenamiento y validación con entrenando con el 70% de los datos.

6. OPTIMIZACIÓN DE FÁRMACOS

	N° NEURONAS														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RMS entr	0.1 22 84	0.1 00 08	0.0 85 46	0.0 66 68	0.0 53 39	0.0 47 20	0. 03 54	0.0 27 54	0.0 17 93	0.0 11 36	0.0 10 00	0.0 09 98	0.0 09 97	0.0 09 99	0.0 09 97
RMS valid	0.2 18 41 5	0.2 15 83 7	0.2 19 03 8	0.3 07 43 9	0.1 94 99 4	0.2 93 90 8	0. 24 96 17	0.2 16 34 0	0.2 86 26 2	0.1 84 10 1	0.1 92 28 6	0.1 93 22 5	0.1 99 09 0	0.1 88 65 5	0.1 94 62 7
RMS entr	0.1 22 68	0.1 00 31	0.0 85 99	0.0 66 92	0.0 52 41	0.0 44 16	0. 03 49	0.0 27 01	0.0 17 98	0.0 11 46	0.0 10 00	0.0 09 99	0.0 09 99	0.0 09 99	0.0 09 99
RMS valid	0.2 18 44 5	0.2 28 00 9	0.2 06 72 0	0.2 13 17 8	0.2 09 97 3	0.2 53 71 8	0. 23 75 16	0.2 09 36 9	0.2 03 59 1	0.2 11 79 9	0.1 92 47 6	0.1 76 18 1	0.1 88 54 0	0.1 84 47 3	0.1 85 66 6
RMS entr	0.1 22 66	0.1 01 47	0.0 88 03	0.0 65 01	0.0 52 69	0.0 44 94	0. 03 50	0.0 27 65	0.0 17 88	0.0 11 35	0.0 09 99	0.0 09 99	0.0 09 99	0.0 10 00	0.0 09 98
RMS valid	0.2 18 42 9	0.2 11 56 1	0.2 18 38 1	0.2 34 36 9	0.2 39 87 0	0.1 84 40 4	0. 20 13 30	0.2 11 75 6	0.2 55 02 4	0.2 01 29 2	0.1 83 56 7	0.1 90 88 5	0.1 78 66 6	0.1 82 14 8	0.1 77 49 8
RMS entr	0.1 21 92	0.1 00 26	0.0 86 49	0.0 70 33	0.0 57 08	0.0 45 50	0. 03 50	0.0 26 16	0.0 18 22	0.0 13 82	0.0 09 99	0.0 09 99	0.0 09 98	0.0 09 99	0.0 09 99
RMS valid	0.2 11 96 3	0.2 27 99 9	0.2 12 78 9	0.2 34 09 7	0.3 30 11 7	0.2 16 25 8	0. 25 18 15	0.2 40 79 1	0.1 93 71 3	0.1 79 17 8	0.1 85 19 9	0.1 89 52 6	0.1 82 63 4	0.1 91 49 2	0.1 83 97 0
RMS entr	0.1 22 63	0.1 00 26	0.0 82 87	0.0 63 52	0.0 54 33	0.0 44 86	0. 03 73	0.0 31 36	0.0 18 05	0.0 11 36	0.0 09 99	0.0 09 99	0.0 09 98	0.0 09 98	0.0 09 98
RMS valid	0.2 18 37 0	0.2 27 72 5	0.2 48 26 1	0.3 20 13 9	0.2 49 66 5	0.3 07 22 0	0. 30 45 62	0.2 17 09 1	0.1 79 87 3	0.1 99 06 6	0.1 87 65 8	0.1 90 29 7	0.1 94 40 9	0.1 79 94 8	0.1 85 72 0
RMS Media	0.2 17 12	0.2 22 23	0.2 21 04	0.2 61 84	0.2 44 92	0.2 51 10	0. 24 89	0.2 19 07	0.2 23 69	0.1 95 09	0.1 88 24	0.1 88 02	0.1 88 67	0.1 85 34	0.1 85 50

Tabla 6.10: Errores en entrenamiento y validación con entrenando con el 80% de los datos.

6. OPTIMIZACIÓN DE FÁRMACOS

	N° NEURONAS														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RMS entr	0.1 28 78	0.1 13 45	0. 08 76	0.0 86 00	0.0 76 20	0.0 50 99	0. 04 73	0.0 34 27	0.0 25 97	0.0 17 19	0. 01 14	0. 02 80	0.0 10 00	0. 00 99	0.0 09 98
RMS valid	0.1 87 93 5	0.1 76 20 4	0. 20 32 70	0.2 08 50 2	0.1 91 42 0	0.1 47 72 1	0. 26 33 31	0.1 70 38 8	0.1 21 19 1	0.1 55 73 5	0. 15 04 26	0. 05 29 94	0.0 69 15 2	0. 09 34 49	0.0 85 22 1
RMS entr	0.1 28 78	0.1 12 61	0. 08 99	0.0 80 12	0.0 67 16	0.0 50 74	0. 04 71	0.0 40 91	0.0 37 34	0.0 37 31	0. 01 13	0. 01 00	0.0 09 99	0. 00 99	0.0 09 99
RMS valid	0.1 87 93 5	0.1 51 69 3	0. 15 04 00	0.3 02 62 4	0.1 94 83 4	0.2 34 16 9	0. 12 94 15	0.1 42 85 0	0.1 96 78 7	0.1 84 76 6	0. 15 42 08	0. 10 12 95	0.0 63 61 1	0. 06 71 06	0.1 00 97 8
RMS entr	0.1 28 78	0.1 12 61	0. 08 47	0.0 73 86	0.0 62 48	0.0 49 93	0. 04 67	0.0 35 43	0.0 27 66	0.0 17 27	0. 01 16	0. 01 00	0.0 09 99	0. 00 99	0.0 09 99
RMS valid	0.1 87 93 5	0.1 51 70 4	0. 14 63 89	0.0 85 48 9	0.1 30 32 6	0.2 67 12 1	0. 17 03 12	0.2 57 36 1	0.2 14 25 8	0.0 99 63 8	0. 11 57 72	0. 08 70 68	0.0 73 85 2	0. 08 28 81	0.0 84 85 9
RMS entr	0.1 28 68	0.1 10 87	0. 10 28	0.0 73 99	0.0 62 38	0.0 51 82	0. 04 21	0.0 35 57	0.0 24 70	0.0 17 14	0. 01 11	0. 01 00	0.0 09 99	0. 00 99	0.0 09 99
RMS valid	0.1 88 03 6	0.1 41 24 2	0. 19 58 04	0.2 13 90 2	0.1 36 58 9	0.2 59 06 9	0. 21 47 07	0.1 92 69 0	0.1 15 45 7	0.1 42 76 1	0. 10 17 65	0. 08 22 99	0.0 97 57 0	0. 08 64 51	0.0 59 05 2
RMS entr	0.1 28 78	0.1 12 61	0. 97 16	0.7 16 69	0.0 63 07	0.0 54 21	0. 04 26	0.0 37 77	0.0 24 36	0.0 17 26	0. 01 11	0. 01 00	0.0 09 99	0. 00 99	0.0 09 99
RMS valid	0.1 87 93 5	0.1 51 69 7	0. 16 37 52	0.1 36 96 6	0.1 12 62 3	0.1 46 33 3	0. 16 90 75	0.3 35 54 2	0.2 21 26 8	0.1 08 83 8	0. 11 56 04	0. 12 36 39	0.1 01 58 6	0. 07 87 08	0.0 85 25 1
RMS Media	0.1 87 96	0.1 54 55	0. 17 19	0.1 89 50	0.1 53 16	0.2 10 88	0. 18 93	0.2 19 77	0.1 73 79	0.1 38 35	0. 12 75	0. 08 04	0.0 81 15	0. 08 17	0.0 83 07

Tabla 6.11: Errores en entrenamiento y validación con entrenando con el 90% de los datos.

En las tablas anteriores cuando el número de neuronas en la capa oculta es menor que 12-13, el error RMS cometido por las redes neuronales en los patrones de entrenamiento es mayor que 0.01. Es decir, el error durante el entrenamiento nunca cae por debajo del error mínimo deseado y el entrenamiento termina en el número máximo de iteraciones permitido. Por tanto, el número de neuronas ocultas para este caso será mayor o igual a 12.

Para este caso se decidirá entre 12 y 15 neuronas. En la siguiente tabla se muestra el número de neuronas donde el error RMS es menor en los patrones de validación para cada caso. Se ha establecido 14 como el número óptimo de neuronas en la capa oculta para esta red neuronal ya que es el mínimo número de neuronas para el caso en que se entrenó con el 80% de los datos y se puede añadir una neurona más a los otros casos sin afectar significativamente al resultado.

% Datos en Entrenamiento	Nº Neuronas (1)	Nº Neuronas (2)
70	13	11
80	14	15
90	12	13

Tabla 6.12: Número óptimo de neuronas ocultas.

Con esto, la arquitectura de red neuronal para este caso será el perceptrón de tres capas con 8 neuronas en la capa de entrada, 14 neuronas en la capa oculta y 16 neuronas en la de salida.

6.4.2. Validación de las redes neuronales.

En la fabricación de fármacos, tras realizar experimentos en el laboratorio, se pasa a realizar un análisis de optimización donde se busca el nivel óptimo de unas características del fármaco para obtener valores adecuados de otras características o

respuestas de dicho fármaco. Uno de los métodos empleados para realizar el análisis de optimización donde existe múltiples respuestas es la regresión. Dicho método implica una gran cantidad de cálculos.

Se propuso abordar el problema de optimización empleando redes neuronales. En este apartado se comprobará el comportamiento de dichas redes neuronales. Se ha planteado anteriormente el problema de entrenar y validar redes neuronales con un número pequeño de datos, como es el caso que estamos tratando, y se propuso entrenar la red neuronal con todos los datos y validar dicha red neuronal empleando el método estadístico de remuestreo con reposición *Bootstrap*. El método de validación de la red neuronal con patrones de validación independientes de los patrones de entrenamiento se comparará con este método.

Se repetirá la misma metodología de comparación de métodos para las dos muestras de datos que disponemos.

Datos I:

Se ha evaluado el comportamiento de la red neuronal utilizando los datos de la Facultad de Farmacia de Cluj-Napoca. La arquitectura empleada para ello es el perceptrón multicapa con 5 neuronas de entrada, 6 neuronas de salida y 7 neuronas en la capa oculta. En el apartado anterior se entrenó varias redes neuronales con distintos números de neuronas en la capa oculta con el objetivo de determinar su número óptimo. Para este problema se llegó a la conclusión que 7 era dicho número óptimo de neuronas ocultas.

En la Tabla 6.13 se ha entrenado una red neuronal varias veces con la misma inicialización. Se ha entrenado 5 veces variando los datos de entrenamiento utilizando el 70% de los datos y otras 5 veces con el 80%.

Los parámetros que son necesarios definir para el entrenamiento son los mismos que en el apartado anterior cuando se determinó el número óptimo de neuronas ocultas. Estos son: tasa de aprendizaje de valor 0.1 y momento de valor 0.9. La condición de terminación del entrenamiento es cuando el error RMS cae por debajo de 0.01, cuando

el entrenamiento realiza 10.000 iteraciones o cuando el gradiente mínimo alcanza el valor de 0.00001.

Este método de verificar el error cometido por la red neuronal empleando patrones de validación independientes de los patrones de entrenamiento se ha comparado con el método de remuestreo *Bootstrap*.

Para estimar el error que comete la red neuronal mediante este método se entrena la red con todos los datos, en este caso 11 patrones. Tras el entrenamiento se evalúa el comportamiento de la red. Para ello se formaron 5000 muestras a partir de los patrones de entrenamiento. Estas muestras se tomaron muestreando los patrones de entrenamiento con reposición y contienen 11 patrones cada una. Con cada una de estas 5000 muestras se halló el error cometido por la red y con la colección de los 5000 errores se halló el error RMS media cometido por la red neuronal junto con su error estándar SE y el sesgo cometido o bias.

Los resultados se muestran en la siguiente tabla:

RED	RMS entrenamiento	RMS validación
1 (70%)	0.009995	0.098368
2	0.009999	0.049662
3	0.009999	0.030285
4	0.009996	0.091652
5	0.010000	0.099092
Media	0.0099978	0.073812
6(80%)	0.009995	0.044275
7	0.009999	0.040615
8	0.010000	0.037153
9	0.009999	0.101211
10	0.009998	0.051837
Media	0.009998	0.055018
BOOTSRAAP	RMS real 0.001100	SE 0.000140 BIAS 0.000997

Tabla 6.13: Resultados de la validación de la red neuronal.

Como se puede observar en la tabla, el error RMS cometido por las redes neuronales en los patrones de validación varía mucho entre una red y otra. Esto es así debido a que las redes neuronales ofrecen buenos resultados cuando el número de patrones de entrenamiento y validación son muy grandes. En casos donde el número de patrones es muy pequeño, los resultados pueden variar enormemente según la elección de los datos utilizados para el entrenamiento. Además se está validando la red neuronal con sólo dos o tres datos cuyo error no se puede afirmar que sea la que realmente cometerá la red neuronal.

Cuando se entrena la red neuronal con todos los datos y se valida con el método de bootstrap, el error RMS obtenido es 0.0011 y el sesgo cometido es 0.001 Este sesgo no es despreciable ya que mayor que la cuarta parte del SE. Este sesgo es de esperar ya que tres de los patrones pertenecen a un experimento repetido tres veces en el laboratorio para obtener resultados distintos. Con esto el error que en realidad comete la red neuronal es $0.0011+0.001=0.0021$.

Datos II:

Se ha procedido de la misma manera que con los datos anteriores para determinar la utilidad de las redes neuronales en la optimización de fármacos utilizando los datos procedentes de la Facultad de Farmacia de Sevilla. Los resultados se muestran en la Tabla 6.14. Los primeros resultados son de haber entrenado dividiendo los datos en patrones de entrenamiento (70% y 80% de los datos) y validación. Al final de la tabla se muestran los resultados obtenidos de haber entrenado con todos los patrones y validado la red mediante el método de Bootstrap. En el método de Bootstrap, al igual que con los datos anteriores, se formó 5000 muestras aleatorias de tamaño 18 con reposición a partir de los datos obtenidos mediante DOE. Para cada una de estas muestras se halló el error RMS cometido por la red neuronal y se halló la media de los 5000 valores RMS. A este error RMS se calculó el sesgo o bias y el error estándar.

RED	RMS entrenamiento	RMS validación
1 (70%)	0.009968	0.193264
2	0.009993	0.220282
3	0.009994	0.232512
4	0.009977	0.224929
5	0.009991	0.240933
Media	0.009985	0.222384
<hr/>		
6(80%)	0.009995	0.188655
7	0.009996	0.257797
8	0.009984	0.214918
9	0.009997	0.232989
10	0.009996	0.257200
Media	0.009994	0.230318
<hr/>		
BOOTSRAAP	RMS real 0.001800	SE 0.000572 BIAS 0.001706

Tabla 6.14: Resultados de la validación de la red neuronal.

Al igual que en el caso anterior, el error cometido por las redes neuronales en los patrones de validación varían entre una red y otra. Al validar la red neuronal con el método de bootstrap, el error RMS obtenido es 0.018 y el sesgo cometido es 0.0017. Este sesgo es mayor que la cuarta parte del SE, por lo que el error RMS de la red neuronal será $0.0018+0.0017=0.0035$.

Si los datos con los que se entrenó la red neuronal es una buena muestra representativa de la población, este error calculado mediante bootstrap debería ser el error real de la red neuronal.

6.4.3. Optimización

La Facultad de farmacia de la Universidad de Sevilla, además de los resultados de los experimentos DOE, nos han facilitado los resultados de la optimización que realizaron mediante el método de regresión. En este apartado compararemos la optimización de este fármaco mediante esta técnica con la optimización que se obtiene empleando las redes neuronales.

En la optimización de un fármaco se busca los valores de las variables de entrada que minimice el valor de una variable de salida, que la maximice o que se aproxime a un valor dado. Los investigadores disponen de varias técnicas entre las que elegir, cada una con sus ventajas e inconvenientes. Un método empleado cuando el sistema tiene respuestas múltiples es realizar una regresión con los datos obtenidos previamente mediante el DOE. Sin embargo, este método presenta el inconveniente de necesitar un gran número de cálculos.

En la Tabla 6.1 se muestran los resultados obtenidos de la optimización del fármaco mediante regresión lineal. Las dos primeras columnas indican la respuesta optimizada y la tercera columna es la solución a la optimización, esto es, las entradas que optimizan respuestas indicadas en las dos primeras columnas. La cuarta columna son los valores medios junto con un intervalo de confianza.

La primera respuesta se ha optimizado tres veces. En esta respuesta se busca que su valor sea próximo a 120. Esto se consigue con el método empleando a partir de la respuesta máxima y la mínima que ha tenido que ser hallado previamente.

RESPUESTA N°	RESPUESTA	L. Opt. Individual	Respuesta IC ($\alpha=0.005$)
1	$t_{50\% \text{ máx.}}$	$A_1B_1C_1D_1E_0F_{-1}G_0H_{-1}$	253.33 ± 46.08
1	$t_{50\% \text{ min.}}$	$A_{-1}B_{-1}C_{-1}D_0E_1F_0G_0H_1$	109.24 ± 2.10
1	$t_{50\% \text{ min.}}$ dist.a a 120	$A_1B_0C_1D_0E_1F_0G_{-1}H_1$	122.64 ± 9.07
2	% ₄₂₀	$A_{-1}B_{-1}C_0D_0E_{-1}F_{-1}G_0H_1$	91.41 ± 4.19
3	EE	$A_1B_0C_0D_0E_1F_0G_1H_{-1}$	85.55 ± 2.62
4	HTO	$A_1B_{-1}C_{-1}D_0E_0F_0G_{-1}H_1$	3073.33 ± 138.28
5	Ds	$A_{-1}B_{-1}C_1D_0E_1F_0G_0H_{-1}$	24.77 ± 0.50
6	AUC	$A_{-1}B_{-1}C_0D_0E_1F_0G_0H_1$	24350.77 ± 694.3
7	THTO _{Máx}	$A_{-1}B_1C_1D_1E_{-1}F_1G_1H_0$	140.00 ± 39.20
8	RA	$A_{-1}B_{-1}C_1D_{-1}E_0F_{-1}G_0H_0$	1.5 ± 0.30
9	C	$A_{-1}B_0C_1D_{-1}E_0F_{-1}G_{-1}H_0$	1.37 ± 0.20
10	DF	$A_{-1}B_1C_1D_0E_0F_{-1}G_{-1}H_1$	1.03 ± 0.023
11	HG	$A_1B_1C_1D_1E_1F_0G_0H_0$	0.42 ± 0.09
12	FF	$A_{-1}B_0C_1D_{-1}E_0F_{-1}G_{-1}H_0$	2.36 ± 0.64
13	A	$A_{-1}B_{-1}C_0D_{-1}E_0F_{-1}G_{-1}H_1$	1.53 ± 0.350
14	Dm	$A_{-1}B_{-1}C_0D_{-1}E_0F_{-1}G_{-1}H_1$	1.37 ± 0.16
15	P	$A_{-1}B_0C_1D_{-1}E_0F_{-1}G_{-1}H_1$	6.64 ± 0.85
16	Dfe	$A_{-1}B_0C_{-1}D_{-1}E_0F_{-1}G_{-1}H_1$	1.8 ± 0.29

Tabla 6.15: Optimización mediante regresión lineal.

Generalmente se trabaja con una codificación de los datos y no con sus valores reales. Esta codificación se muestra en la Tabla 6.16.

FACTOR	Niveles		
	1	2	3
A. Cantidad de TEA (gotas)	0	4	-
B. Concentración de Carbopol (%)	1	2	4
C. Velocidad de flujo de caída (mL/min)	1	2	3
D. Velocidad de agitación (r.p.m.)	100	300	500
E. Tiempo de reacción con los iones Ca ²⁺ (min.)	5	30	60
F. Secado	<i>Ambiente</i>	<i>Rotavapor</i>	<i>Estufa</i>
G. Adición de electrolito (NaCl M)	0	0.01	0.1
H. pH del medio	7.2	7.6	8

Tabla 6.16: Codificación de los datos.

Los criterios de optimización para cada salida son los siguientes:

- Maximizar: %420, AUC, EE, HTO, Ds y FF.
- Minimizar: Thto, Dm, A, P, DFe, DF, HG, RA, C
- Aproximar a un valor: $t_{50\%}=120$

El funcionamiento normal de una red neuronal es ofrecer datos de salida para una entrada determinada. Sin embargo, en la optimización de fármacos se busca los valores de entrada para una salida, es decir, lo contrario a la forma de actuar de una red neuronal.

Para resolver este problema se formó todas las combinaciones posibles de los valores de las entradas. Cada una de estas combinaciones se pasó por la red neuronal y

se guardaron todos los resultados. Para obtener las entradas óptimas que minimice el valor de una salida, se busca el mínimo de dicha salida entre todos los resultados guardados anteriormente obteniendo así las entradas buscadas. Se procede de igual manera a la hora de maximizar una salida o aproximarla a un valor deseado.

En la siguiente tabla se muestran el resultado óptimo de las entradas para cada respuesta que se optimizó.

RESPUESTA Nº	RESPUESTA	L. Opt. Individual
1	$t_{50\% \text{ min.}}$ dist.a a 120	$A_{-1}B_1C_1D_{-1}E_{-1}F_{-1}G_0H_{-1}$
2	% 420	$A_{-1}B_{-1}C_{-1}D_1E_1F_1G_1H_{-1}$
3	EE	$A_1B_{-1}C_1D_1E_0F_1G_{-1}H_{-1}$
4	HTO	$A_{-1}B_{-1}C_{-1}D_{-1}E_1F_{-1}G_{-1}H_1$
5	Ds	$A_{-1}B_{-1}C_0D_1E_0F_1G_{-1}H_1$
6	AUC	$A_{-1}B_{-1}C_{-1}D_0E_{-1}F_1G_{-1}H_1$
7	THTO _{Máx}	$A_{-1}B_{-1}C_{-1}D_1E_1F_{-1}G_{-1}H_1$
8	RA	$A_1B_{-1}C_1D_{-1}E_0F_{-1}G_{-1}H_{-1}$
9	C	$A_{-1}B_0C_1D_{-1}E_1F_{-1}G_1H_1$
10	DF	$A_{-1}B_{-1}C_{-1}D_{-1}E_1F_{-1}G_1H_1$
11	HG	$A_{-1}B_1C_1D_{-1}E_0F_{-1}G_{-1}H_{-1}$
12	FF	$A_1B_0C_1D_{-1}E_{-1}F_{-1}G_1H_1$
13	A	$A_{-1}B_{-1}C_1D_0E_{-1}F_1G_{-1}H_1$
14	Dm	$A_1B_1C_{-1}D_{-1}E_1F_1G_{-1}H_{-1}$
15	P	$A_{-1}B_0C_{-1}D_{-1}E_{-1}F_{-1}G_1H_0$
16	Dfe	$A_{-1}B_0C_0D_0E_1F_{-1}G_1H_{-1}$

Tabla 6.17: Optimización empleando redes neuronales.

Vemos que la solución dada mediante regresión lineal difiere del resultado de la optimización obtenida utilizando redes neuronales. En la siguiente tabla se muestran las respuestas predichas por ambos métodos cuando las entradas son las dadas en las tablas anteriores.

RESPUESTA Nº	RESPUESTA	Respuesta predicha según optimización por regresión	Respuesta predicha según optimización con RN
1	t _{50%} min. dist.a a 120	123.61	119.99
2	% ₄₂₀	87.68	100.97
3	EE	87.28	91.82
4	HTO	2527.03	3861.39
5	Ds	24.00	31.30
6	AUC	26086.38	30014.57
7	THTO _{Máx}	205.00	379.74
8	RA	1.45	0.97
9	C	2.52	0.68
10	DF	1.13	1.04
11	HG	0.44	0.10
12	FF	0.42	0.71
13	A	1.80	1.10
14	Dm	1.55	1.25
15	P	6.76	4.68
16	Dfe	1.87	1.37

Tabla 6.18: Respuestas predichas en la optimización.

Como se observa en la tabla anterior, utilizando una red neuronal en la optimización de una salida se obtiene mejores resultados que empleando el método de regresión lineal.

La diferencia en los resultados obtenidos mediante regresión lineal y mediante el empleo de redes neuronales se debe a que en las redes neuronales tanto una neurona como el conjunto de ellas son no lineales y en la mayoría de los casos los datos presentados a la red neuronal también son de carácter no lineal. Esta no linealidad en las redes neuronales permite que hallen relaciones complejas entre los datos de entrada y los de salida a diferencia del método de regresión lineal donde las relaciones buscadas son lineales.

Para terminar el estudio de la utilidad de las redes neuronales para la optimización de fármacos, haría falta una fase de realización de experimentos confirmatorios con el objeto de ver la bondad de las predicciones. Sin embargo, dicha fase no entra dentro del objeto del proyecto pudiendo ser una futura línea de trabajo.