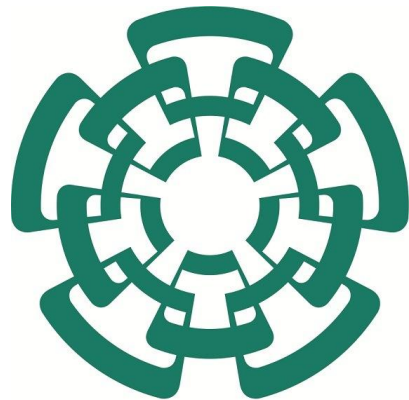


CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS  
AVANZADOS DEL INSTITUTO POLITECNICO  
NACIONAL



**Cinvestav**

Manufactura II

TAREA 3

---

## Algoritmo Backpropagation

---

***Autor:***

Luis Ángel Torres Martínez (214520012)

***Profesor(a):***

Dr. Ismael López Juárez

Saltillo, Coahuila, México  
07 de Septiembre de 2022  
ultima edición: 8 de octubre de 2022

# Índice

<b>1. Resumen</b>	<b>3</b>
<b>2. Introducción</b>	<b>3</b>
<b>3. Desarrollo</b>	<b>3</b>
3.1. Resultados . . . . .	4
3.1.1. Orden 1 . . . . .	4
3.1.2. Orden 2 . . . . .	7
3.1.3. Orden 3 . . . . .	10

## 1. Resumen

En el presente trabajo se mostrarán los resultados del algoritmo de backpropagation para el aprendizaje de la compuerta lógica **And**, **Or** y **Xor**. La red neuronal utilizada consiste en una neurona en la capa de entrada y una neurona en la capa de salida (No hay capas ocultas) y el vector de entrada es de tamaño dos. Además se realiza el entrenamiento de la red con la tabla de verdad de la compuerta Xor en diferente Orden, con el fin de observar el comportamiento en el aprendizaje de la red.

## 2. Introducción

Una red neuronal está compuesta por varios perceptrones y su salida sigue las mismas reglas que el perceptrón, es decir, se realiza la suma ponderada de las entradas y después esta se hace pasar por una función de activación. Esta operación se realiza de manera recursiva de manera que, la entrada de la capa subsecuente corresponde a la salida de la capa actual. De esta manera, mediante la interconexión de varios perceptrones es posible generar funciones no lineales.

La manera en que aprenden las redes neuronales es mediante la minimización de una función de costo, como puede ser el error cuadrático medio (MSE), ReLu, entre otras. Una vez que se tiene la función de costo, se utiliza la técnica del descenso del gradiente para minimizar esta función y mediante esta técnica la red neuronal “aprende”. Como técnica utilizada para obtener el gradiente se utiliza el algoritmo de **backpropagation**.

## 3. Desarrollo

Se implementó una red neuronal que consiste en 1 capa de entrada y 1 capa de salida, cada capa contiene 1 sola neurona, el vector de entrada es de tamaño 2 y la red tiene una topología *fully connected*.

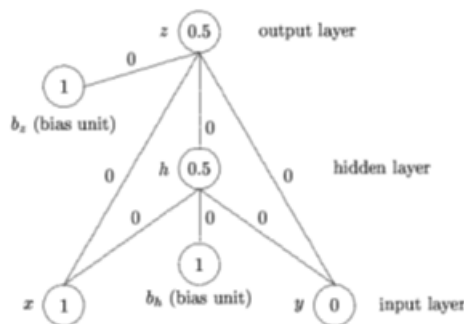


Figura 1: Topología de la red

Se realizó el entrenamiento de las 3 redes neuronales con diferentes ordenes dados en el “dataset” de entrenamiento, diferentes tasas de aprendizaje y diferentes tipos de batch.

### Tasas de aprendizaje:

0.1 - 0.5 - 1 - 2 - 3 - 4 - 5

### Tamaños de lote:

1 - 2 - 4

El tamaño del batch 1 significa entrenamiento en línea, el batch de 4 es el entrenamiento por lote y el batch 2 es entrenamiento de medio lote.

### 3.1. Resultados

#### 3.1.1. Orden 1

In	And	Or	Xor
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

El entrenamiento de la red se realizó con los siguientes resultados:

$\eta$	Epocas								
	En línea			Batch (4)			Batch (2)		
	And	Or	Xor	And	Or	Xor	And	Or	Xor
0.1	13253	11276	22990	25183	36909	más de 50000	14884	22248	más de 50000
0.5	1342	1666	6592	5386	6992	19240	5330	4383	10893
1	672	1141	3332	5313	4403	10176	1460	2308	4610
2	638	465	1518	2648	1896	4608	638	944	4762
3	382	386	más de 50000	905	1486	5237	875	613	1560
4	152	283	más de 50000	1327	1101	2520	325	552	1678
5	241	226	más de 50000	1063	884	1838	297	398	942

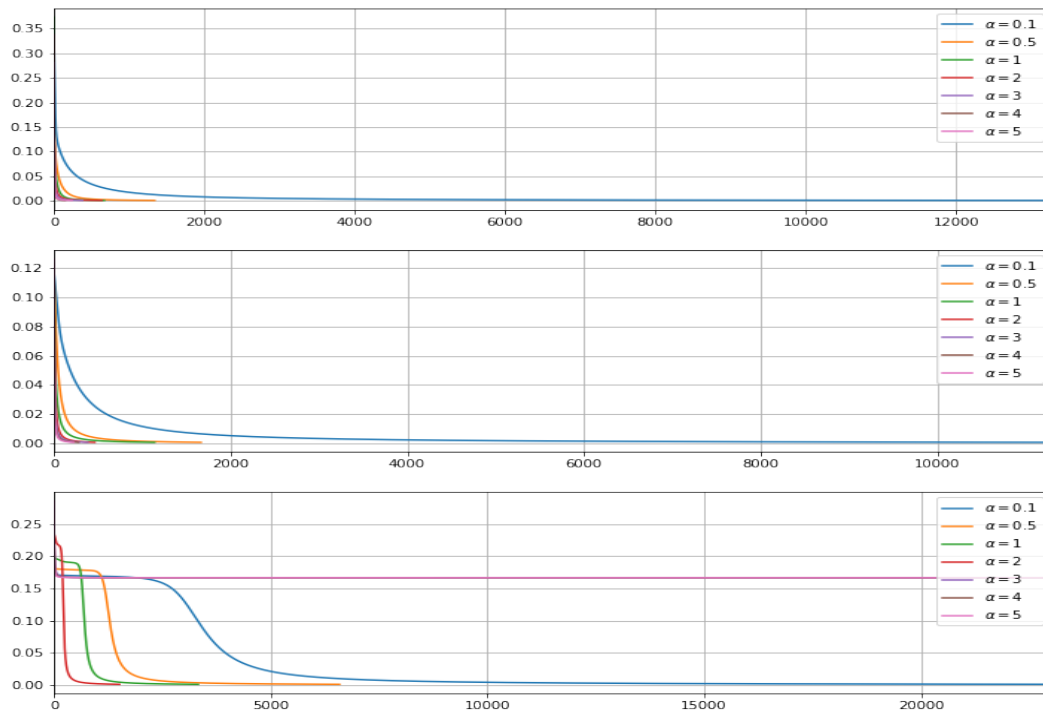


Figura 2: Entrenamiento de las redes con batch 1 (De arriba a abajo: And,Or,Xor)

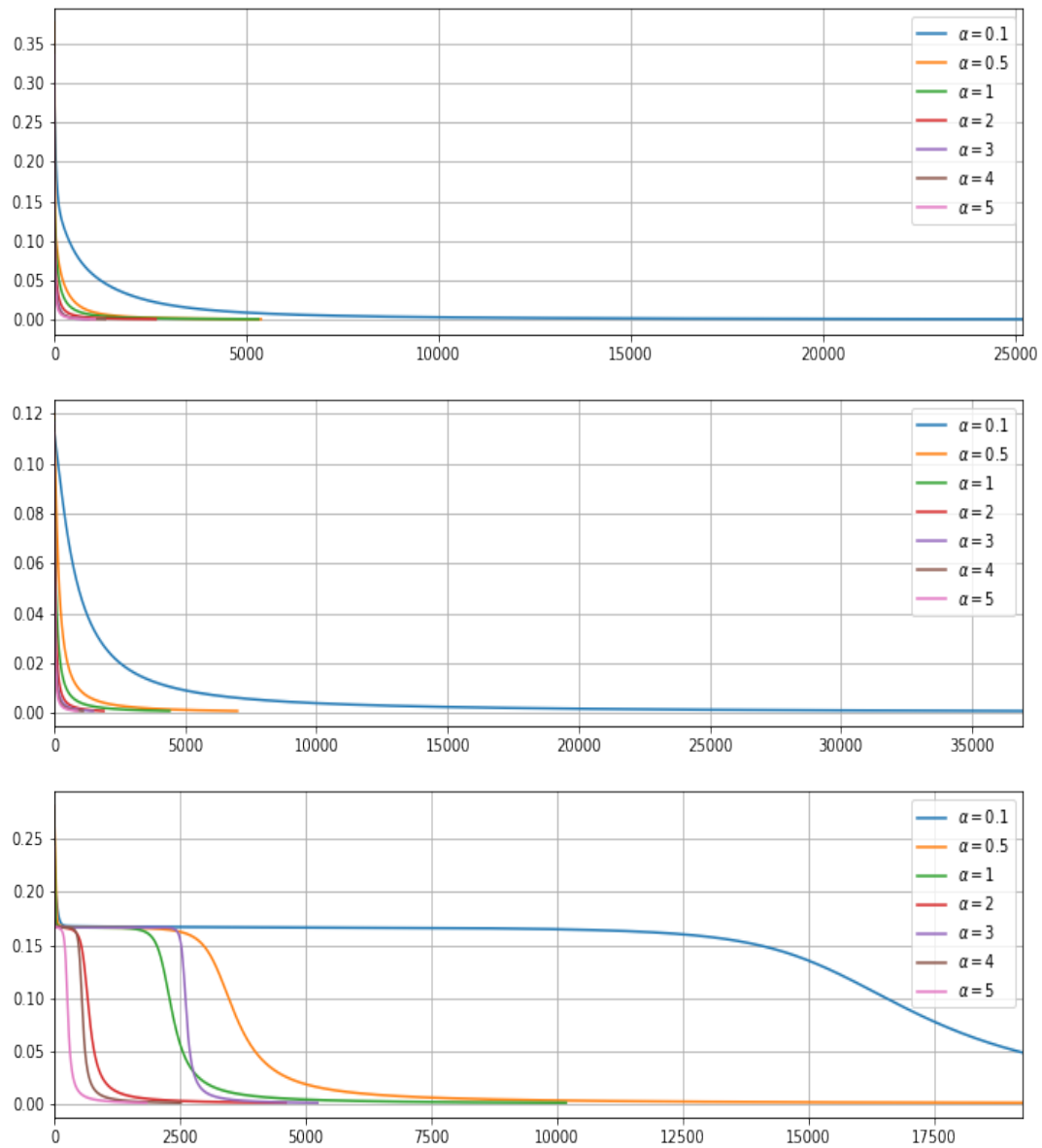


Figura 3: Entrenamiento de las redes con batch 4 (De arriba a abajo: And,Or,Xor)

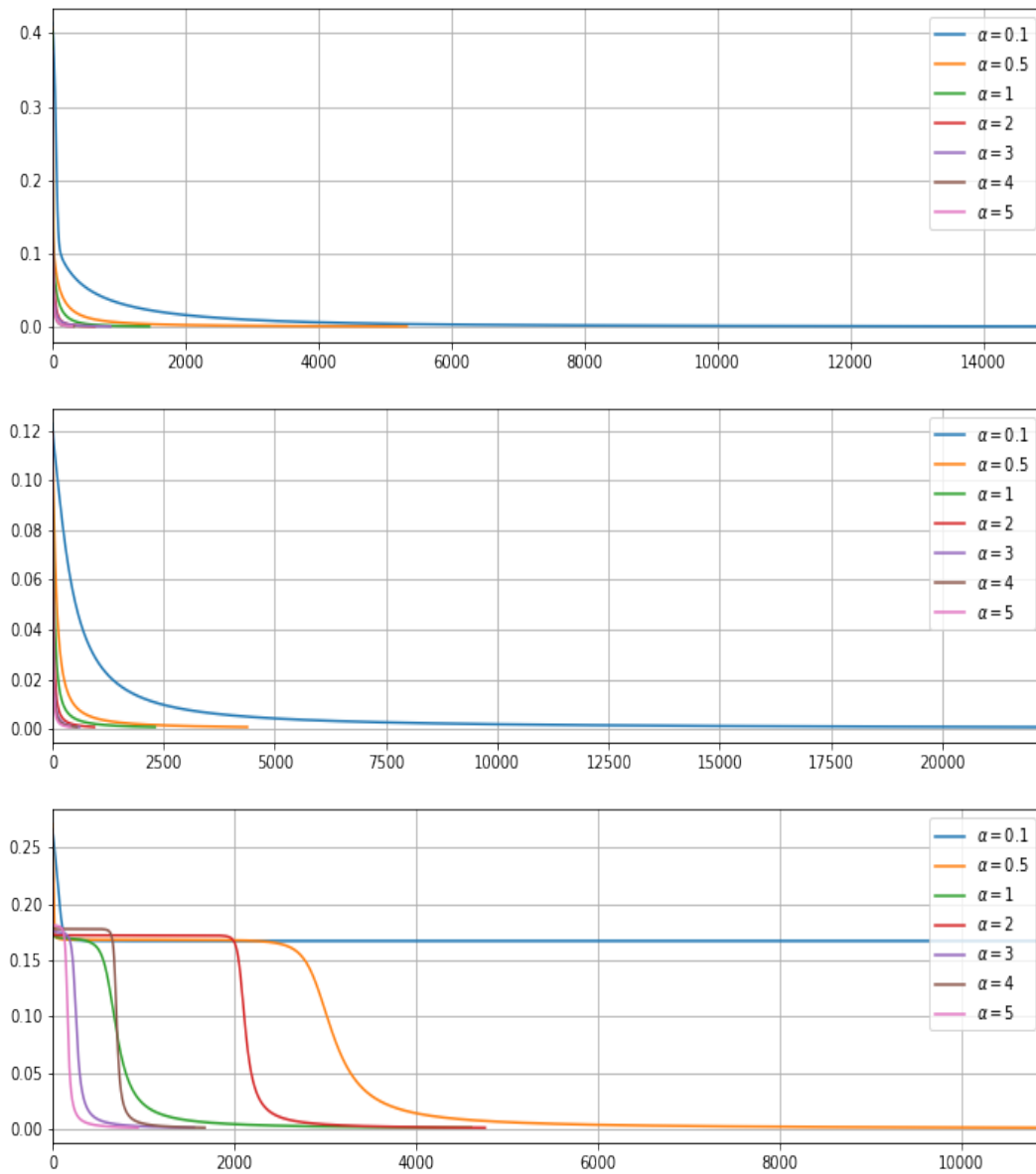


Figura 4: Entrenamiento de las redes con batch 2 (De arriba a abajo: And,Or,Xor)

### 3.1.2. Orden 2

In	And	Or	Xor
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	0

El entrenamiento de la red se realizó con los siguientes resultados:

$\eta$	Epocas								
	En línea			Batch (4)			Batch (2)		
	And	Or	Xor	And	Or	Xor	And	Or	Xor
0.1	6620	10204	24852	45865	46051	más de 50000	26595	22632	más de 50000
0.5	1433	2249	6338	5284	8779	más de 50000	3123	4169	17154
1	1307	1160	2764	5304	4550	9750	2644	2232	4943
2	312	523	1718	1354	2177	5330	1245	1127	7881
3	216	317	851	1770	1523	2948	879	688	1694
4	170	289	más de 50000	1325	974	más de 50000	318	540	1272
5	134	194	más de 50000	1062	824	1782	364	415	más de 50000

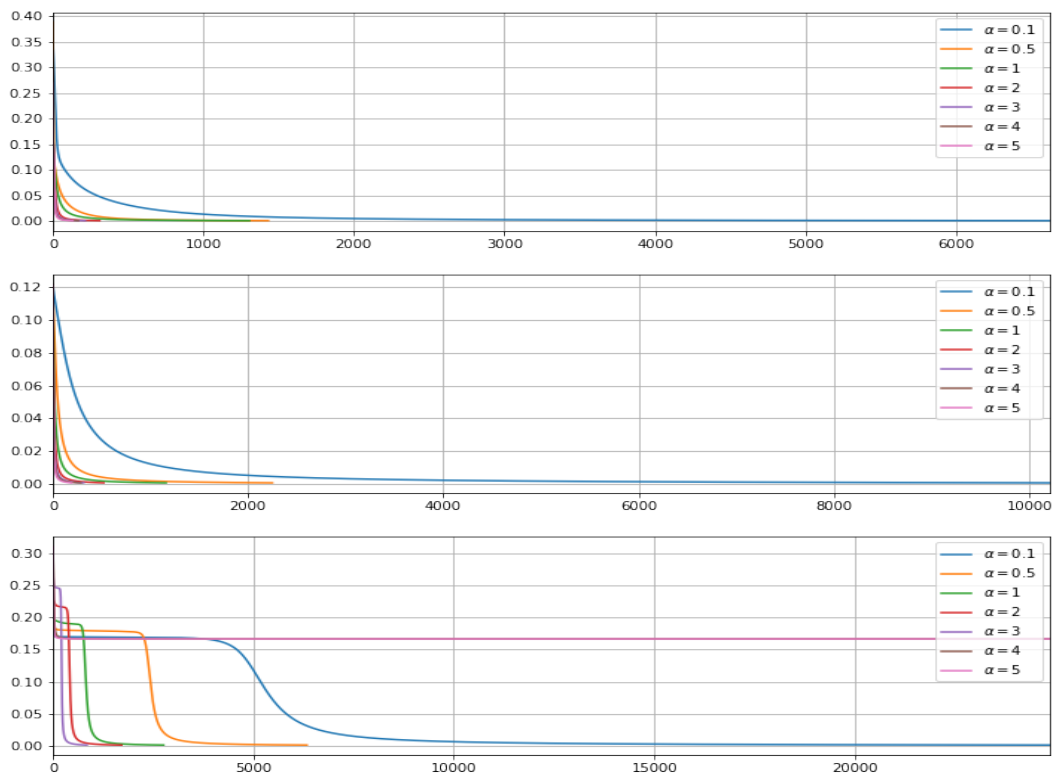


Figura 5: Entrenamiento de las redes con batch 1 (De arriba a abajo: And,Or,Xor)

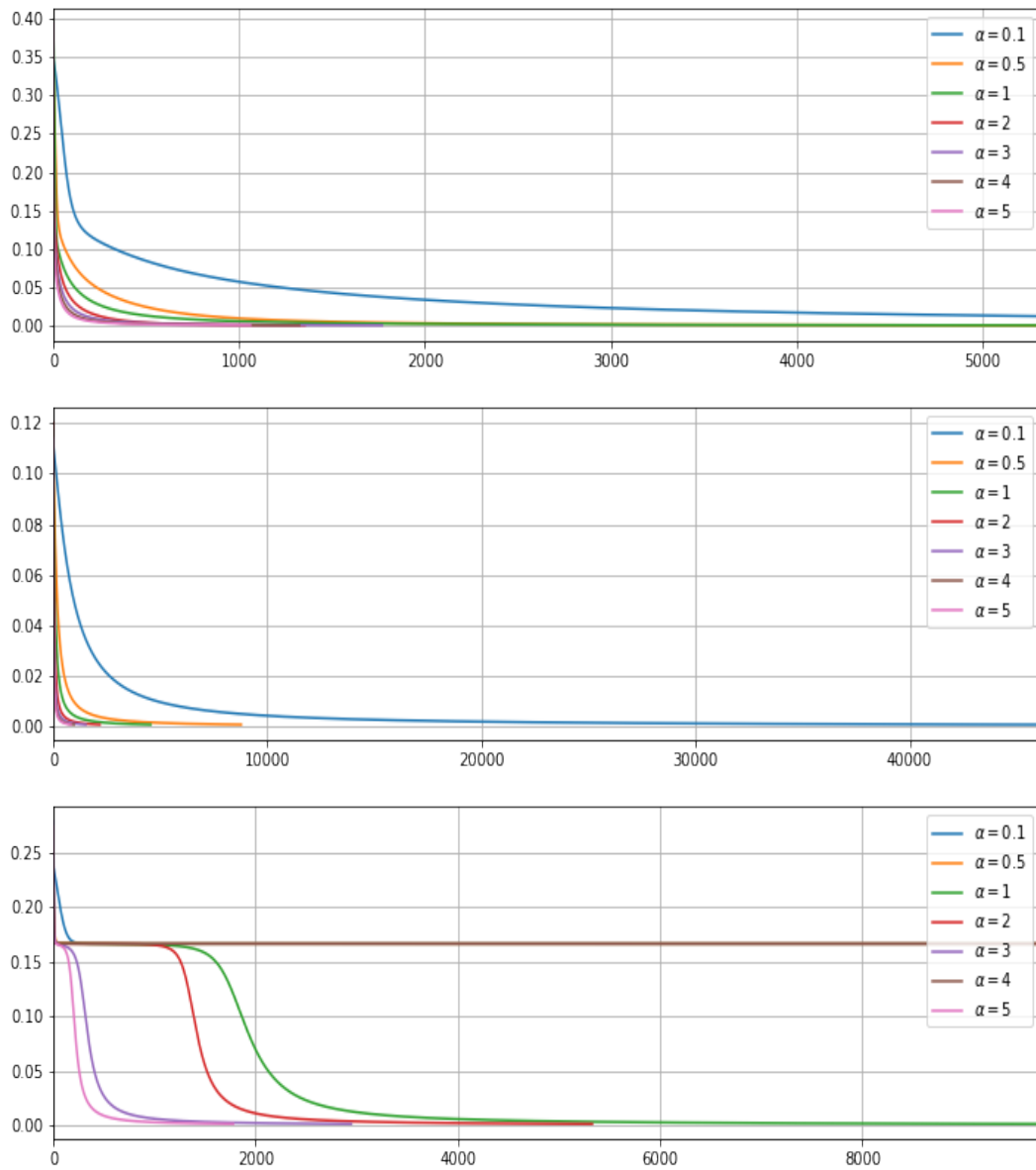


Figura 6: Entrenamiento de las redes con batch 4 (De arriba a abajo: And,Or,Xor)



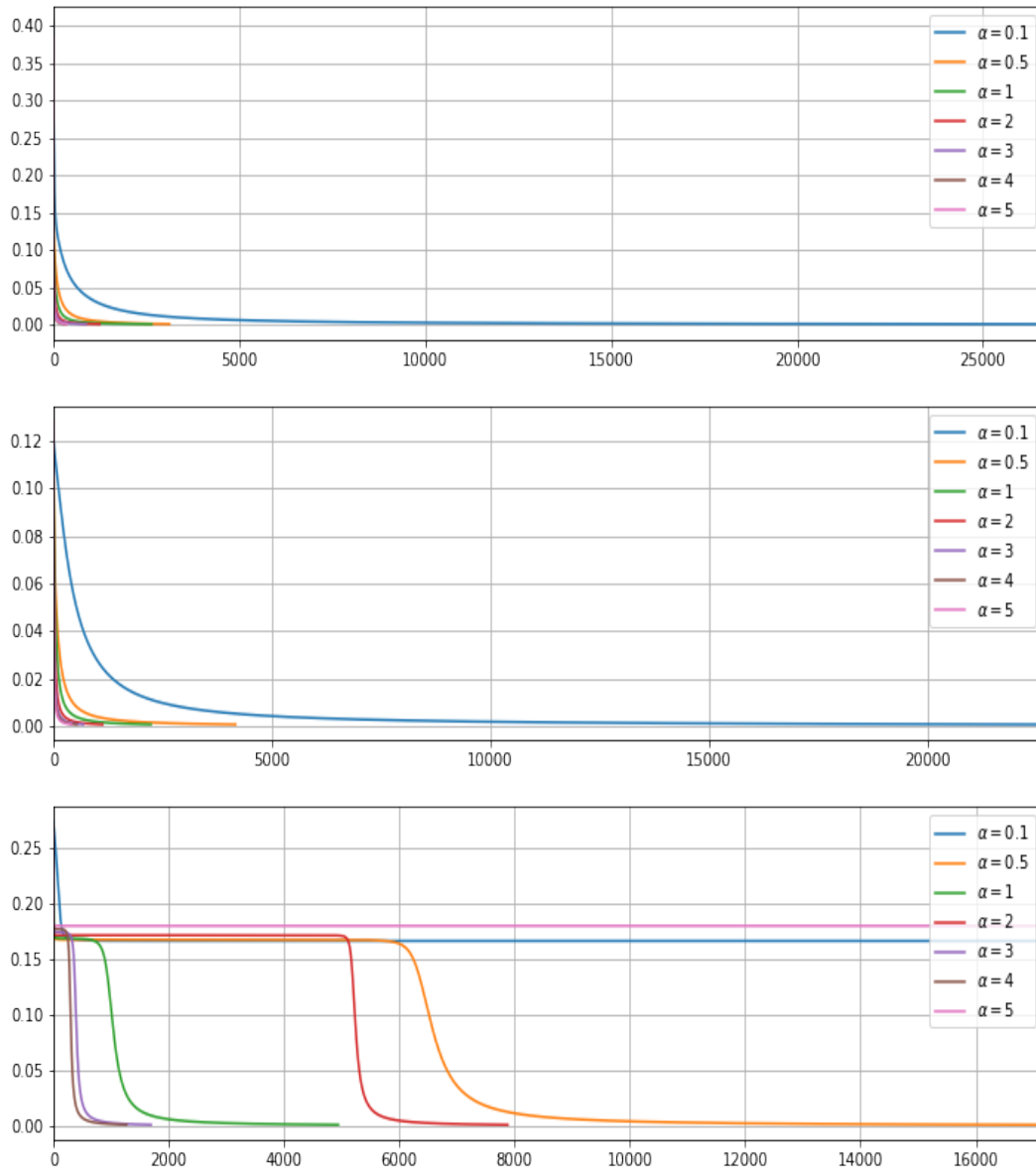


Figura 7: Entrenamiento de las redes con batch 2 (De arriba a abajo: And,Or,Xor)

### 3.1.3. Orden 3

In	And	Or	Xor
0	0	0	0
1	1	1	0
0	1	0	1
1	0	0	1

El entrenamiento de la red se realizó con los siguientes resultados:

$\eta$	Epocas								
	En línea			Batch (4)			Batch (2)		
	And	Or	Xor	And	Or	Xor	And	Or	Xor
0.1	13302	10281	37423	27350	45932	más de 50000	26483	22010	más de 50000
0.5	1416	2176	4403	10607	7013	23004	5288	3889	21725
1	727	1117	2647	5298	4066	9516	1332	2243	7101
2	335	531	1262	2659	2238	más de 50000	1302	1119	2207
3	217	353	801	962	1545	más de 50000	860	767	2055
4	309	250	más de 50000	1297	909	2321	328	432	1654
5	244	221	más de 50000	652	872	1765	260	446	más de 50000

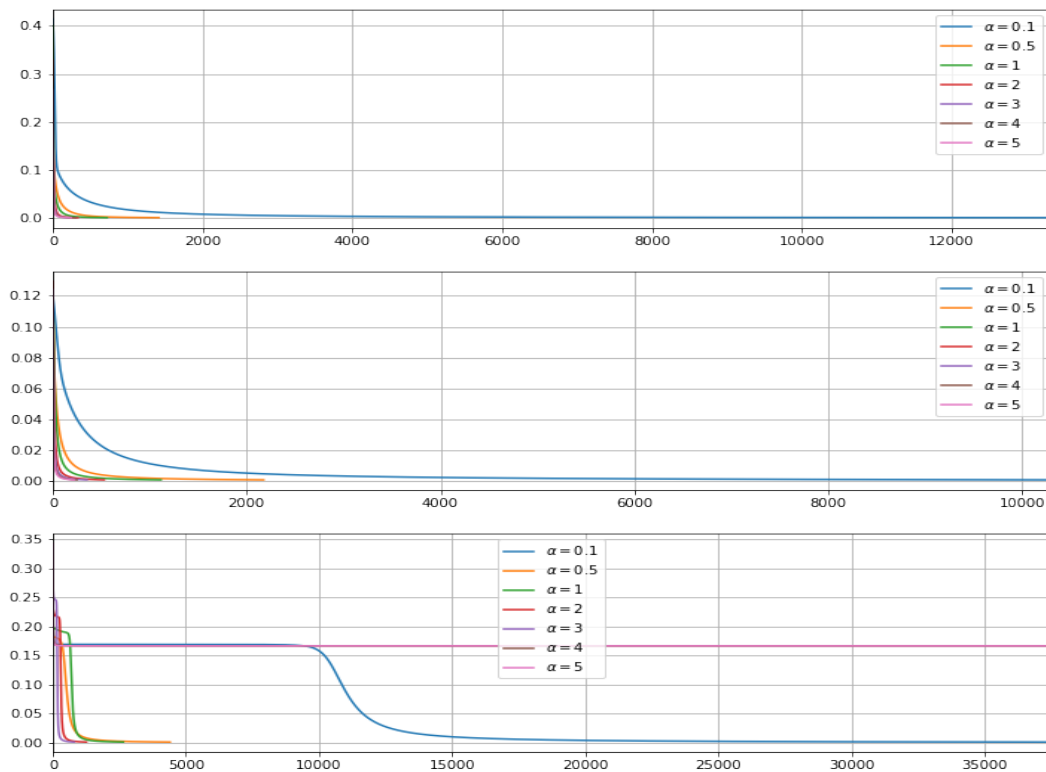


Figura 8: Entrenamiento de las redes con batch 1 (De arriba a abajo: And,Or,Xor)

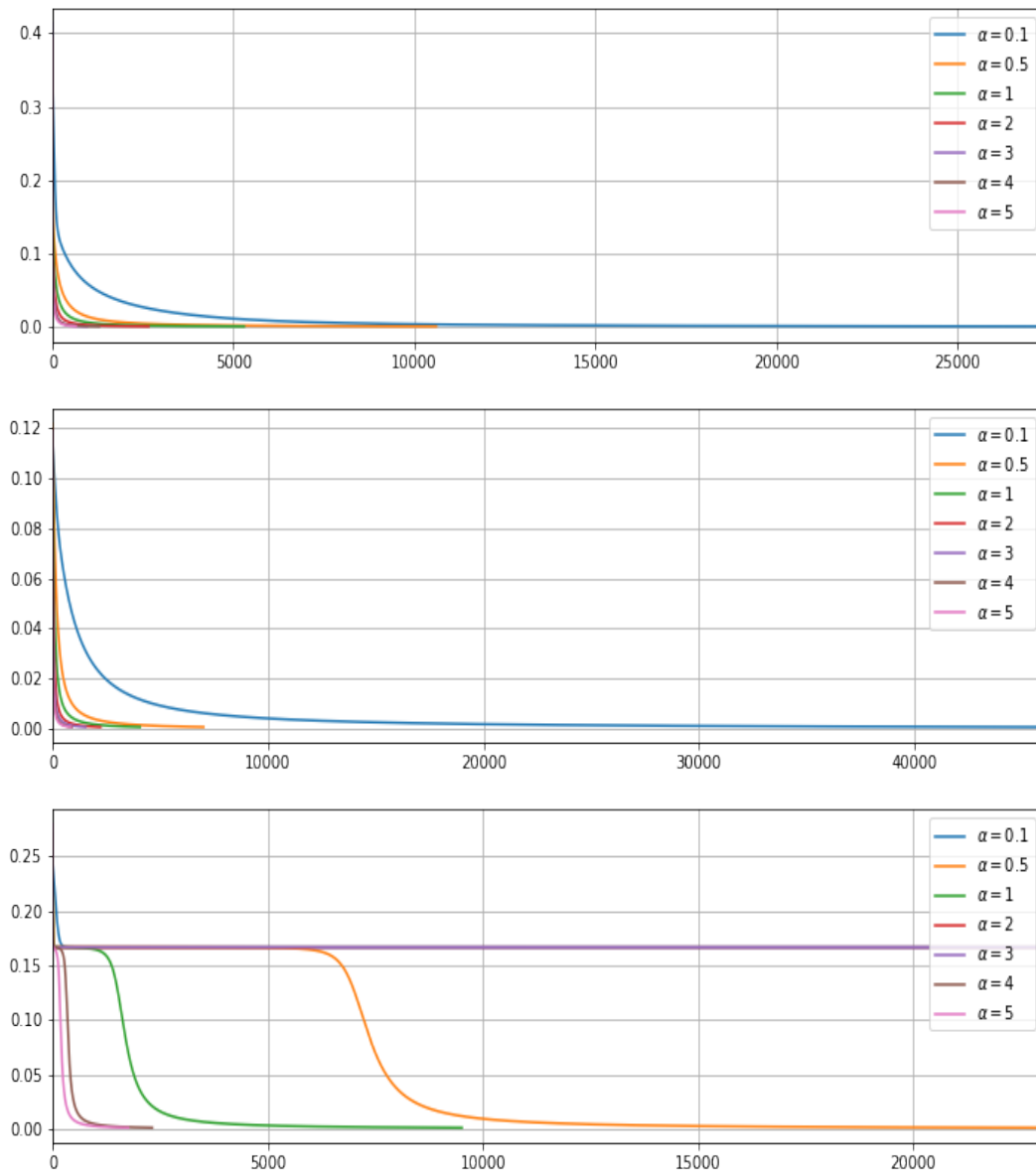


Figura 9: Entrenamiento de las redes con batch 4 (De arriba a abajo: And,Or,Xor)

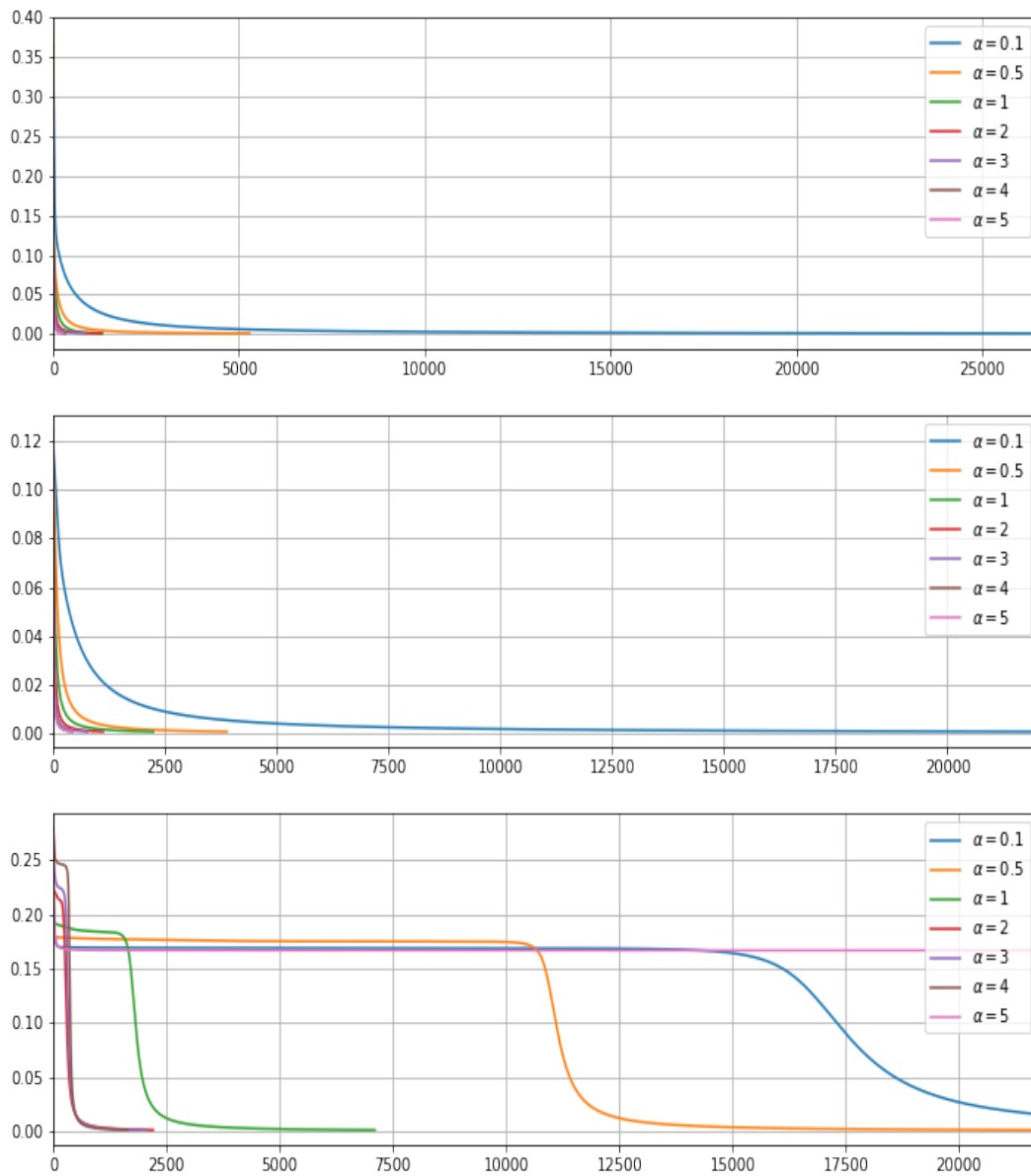


Figura 10: Entrenamiento de las redes con batch 2 (De arriba a abajo: And,Or,Xor)