

## Práctica Tema 8: Deep Learning Cifer 10

El objetivo de esta práctica es experimentar con una *Convolutional Neural Network* (CNN) y realizar un análisis comparativo del rendimiento variando un par de parámetros en la función de entrenamiento sobre un subconjunto de los datos CIFAR-10.

### Descripción del conjunto de datos empleado

El subconjunto de datos escogido está basado en el CIFAR-10 original pero quitando las imágenes relativas a las clases *cat* y *deer*, reduciendo a 8 los tipos de clases con los que se trabajarán:

- *Airplane*
- *Automobile*
- *Bird*
- *Dog*
- *Frog*
- *Horse*
- *Ship*
- *Truck*

El conjunto de datos empleado está formado por dos subconjuntos diferenciados: *test* y *training*. En los datos utilizados para entrenar la red nos encontramos con 8 clases diferenciadas compuestas de 5000 imágenes cada una; mientras que en el conjunto de test, cada una de las clases, dispone de 1000 imágenes.

Las imágenes representan objetos de cada una de las clases con diferentes características (i.e. diferentes tipos de aviones, diferentes colores de coches, diferentes razas de perro, ...) y/o ángulos de cámara (i.e. solo la cara del perro, un rana desde arriba, ...) en un tamaño de 32 x 32 píxeles. Se muestran a continuación algunos ejemplos de cada una de las clases utilizadas:



Teóricamente, las clases son completamente excluyentes entre sí; por ejemplo, no hay superposición entre *automobiles* y *trucks* o entre *dogs* y *cats*.

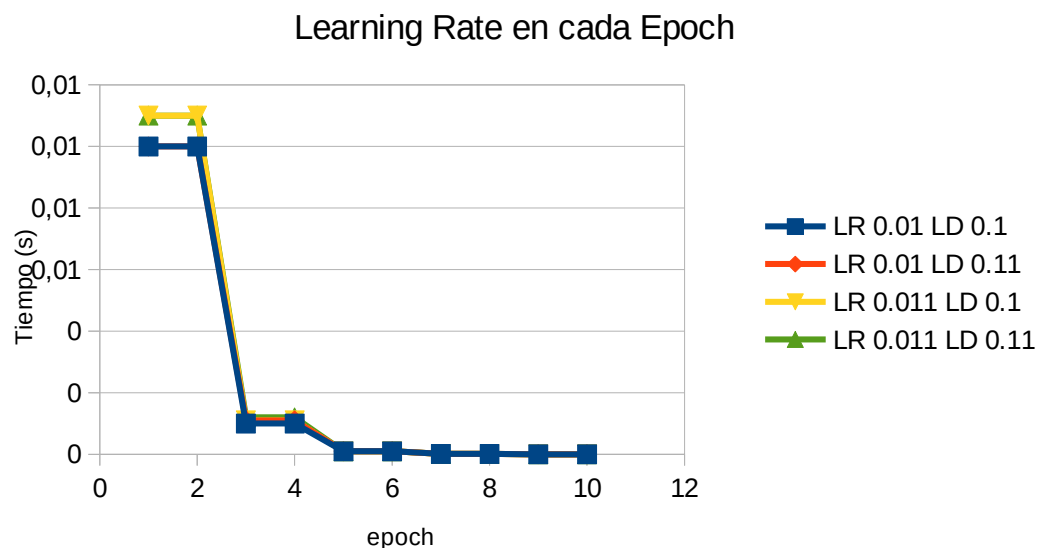
### Estudio sobre el ajuste de los parámetros

Para realizar el estudio de la red neuronal variando los parámetros *InitialLearnRate* y *LearnRateDropFactor* de la función *trainNetwork* de entrenamiento de la red se escogieron los siguientes valores (valores cercanos a los presentados en los scripts del campo virtual para no desestabilizar la red):

<i>InitialLearnRate</i>	0.01 y 0.011
<i>LearnRateDropFactor</i>	0.1 y 0.11

Se ha escogido el valor de la variable *LearnRateDropFactor*, porque de los dos scripts presentes en el campus virtual, el script *cifar10CNN.m* presentaba mejores resultados que el *cifar10CNN\_R2016a.m*. Para realizar las diferentes pruebas de valores se modificó el script X añadiendo una función donde pasar el número de iteraciones a realizar por cada combinación de *InitialLearnRate-LearnRateDropFactor* especificados (véase script para más información). En los siguientes subapartados se mostrarán los resultados obtenidos para las diferentes combinaciones de valores prestando especial interés en la matriz de confusión y en los tiempos de entrenamiento<sup>1</sup>.

Antes de empezar, se muestra la gráfica en la que se representa el valor del *learning rate* para los diferentes *epochs* y combinación de valores:



Como se puede apreciar, en las 4 combinaciones los valores del *learning rate* se comportaron de la misma forma. Las menores variaciones se observaron con el *InitialLearnRate* 0.01 y *LearnRateDropFactor* 0.1 y *InitialLearnRate* 0.011 y *LearnRateDropFactor* 0.1. En la siguiente tabla se muestran los resultados numéricos obtenidos:

Epoch	LR 0.01 LD 0.1	LR 0.01 LD 0.11	LR 0.011 LD 0.1	LR 0.011 LD 0.11
1	0,01	0,01	0,011	0,011
2	0,01	0,01	0,011	0,011
3	0,001	0,0011	0,0011	0,0012
4	0,001	0,0011	0,0011	0,0012
5	0,0001	0,0001	0,0001	0,0001
6	0,0001	0,0001	0,0001	0,0001
7	1E-05	1,331E-05	1,10E-05	1,46E-05
8	1E-05	1,331E-05	1,10E-05	1,46E-05
9	1E-06	1,46E-06	1,10E-06	1,61E-06
10	1E-06	1,46E-06	1,10E-06	1,61E-06

<sup>1</sup> se ejecutaron 20 iteraciones y se descartaron los valores anómalos

Por lo tanto, la variación del *Learning Rate* respecto al epoch no debería de ser un indicador crítico a la hora de seleccionar una mejor combinación porque las diferencias son mínimas.

Más información respecto a las salidas obtenidas durante el entrenamiento se pueden encontrar en los archivos *salida\_<iteracion>* dentro de las carpetas correspondientes.

### ***InitialLearnRate 0.01 y LearnRateDropFactor 0.1***

Se muestran a continuación las matrices de confusión obtenidas:

**Confusion Matrix**

Output Class \ Target Class	0	1	2	3	4	5	6	7	8
1	505 6.3%	41 0.5%	102 1.3%	41 0.5%	23 0.3%	55 0.7%	180 2.2%	44 0.5%	51.0% 49.0%
2	50 0.6%	419 5.2%	38 0.5%	20 0.2%	53 0.7%	39 0.5%	103 1.3%	219 2.7%	44.5% 55.5%
3	55 0.7%	24 0.3%	283 3.5%	133 1.7%	107 1.3%	69 0.9%	19 0.2%	20 0.2%	39.9% 60.1%
4	27 0.3%	25 0.3%	212 2.6%	456 5.7%	106 1.3%	132 1.7%	33 0.4%	29 0.4%	44.7% 55.3%
5	29 0.4%	68 0.9%	170 2.1%	139 1.7%	555 6.9%	68 0.9%	20 0.2%	63 0.8%	49.9% 50.1%
6	50 0.6%	40 0.5%	105 1.3%	137 1.7%	75 0.9%	528 6.6%	39 0.5%	61 0.8%	51.0% 49.0%
7	200 2.5%	127 1.6%	32 0.4%	31 0.1%	10 0.1%	21 0.3%	489 6.1%	113 1.4%	47.8% 52.2%
8	84 1.1%	256 3.2%	58 0.7%	43 0.5%	71 0.9%	88 1.1%	117 1.5%	451 5.6%	38.6% 61.4%
	50.5% 49.5%	41.9% 58.1%	28.3% 71.7%	45.6% 54.4%	55.5% 44.5%	52.8% 47.2%	48.9% 51.1%	45.1% 54.9%	46.1% 53.9%

**Confusion Matrix**

Output Class \ Target Class	0	1	2	3	4	5	6	7	8
1	592 7.4%	30 0.4%	100 1.2%	22 0.3%	8 0.1%	29 0.4%	154 1.9%	38 0.5%	60.8% 39.2%
2	43 0.5%	659 8.2%	16 0.2%	9 0.1%	11 0.1%	8 0.1%	79 1.0%	182 2.3%	65.4% 34.6%
3	95 1.2%	16 0.2%	487 6.1%	140 1.8%	90 1.1%	58 0.7%	10 0.1%	15 0.2%	53.5% 46.5%
4	17 0.2%	9 0.1%	138 1.7%	578 7.2%	64 0.8%	131 1.6%	22 0.3%	20 0.2%	59.0% 41.0%
5	17 0.2%	21 0.3%	123 1.5%	89 1.1%	735 9.2%	38 0.5%	8 0.1%	37 0.5%	68.8% 31.2%
6	30 0.4%	17 0.2%	79 1.0%	115 1.4%	37 0.5%	687 8.6%	17 0.2%	39 0.5%	67.3% 32.7%
7	140 1.8%	71 0.9%	23 0.3%	24 0.3%	12 0.1%	16 0.2%	659 8.2%	68 0.9%	65.1% 34.9%
8	66 0.8%	177 2.2%	34 0.4%	23 0.3%	43 0.5%	33 0.4%	51 0.6%	601 7.5%	58.5% 41.5%
	59.2% 40.8%	65.9% 34.1%	48.7% 51.3%	57.8% 42.2%	73.5% 26.5%	68.7% 31.3%	65.9% 34.1%	60.1% 39.9%	62.5% 37.5%

**Confusion Matrix**

Output Class \ Target Class	0	1	2	3	4	5	6	7	8
1	541 6.8%	47 0.6%	111 1.4%	50 0.6%	23 0.3%	55 0.7%	194 2.4%	43 0.5%	50.8% 49.2%
2	63 0.8%	519 6.5%	53 0.7%	26 0.3%	34 0.4%	33 0.4%	105 1.3%	197 2.5%	50.4% 49.6%
3	54 0.7%	21 0.3%	277 3.5%	150 1.9%	76 0.9%	72 0.9%	17 0.2%	27 0.3%	39.9% 60.1%
4	28 0.4%	18 0.2%	188 2.4%	417 5.2%	107 1.3%	118 1.5%	39 0.5%	24 0.3%	44.4% 55.6%
5	30 0.4%	58 0.7%	189 2.4%	140 1.8%	643 8.0%	76 0.9%	20 0.2%	49 0.6%	53.4% 46.6%
6	49 0.6%	24 0.3%	125 1.6%	164 2.1%	54 0.7%	564 7.0%	22 0.3%	53 0.7%	53.5% 46.5%
7	188 2.4%	110 1.4%	35 0.4%	26 0.3%	15 0.2%	22 0.3%	484 6.0%	136 1.7%	47.6% 52.4%
8	47 0.6%	203 2.5%	22 0.3%	27 0.3%	48 0.6%	60 0.8%	119 1.5%	471 5.9%	47.2% 52.8%
	54.1% 45.9%	51.9% 48.1%	27.7% 72.3%	41.7% 58.3%	64.3% 35.7%	56.4% 43.6%	48.4% 51.6%	47.1% 52.9%	48.9% 51.0%

**Confusion Matrix**

Output Class \ Target Class	0	1	2	3	4	5	6	7	8
1	490 6.1%	41 0.5%	86 1.1%	43 0.5%	22 0.3%	57 0.7%	181 2.3%	66 0.8%	49.7% 50.3%
2	61 0.8%	464 5.8%	40 0.5%	34 0.4%	49 0.6%	26 0.3%	126 1.6%	208 2.6%	46.0% 54.0%
3	74 0.9%	24 0.3%	348 4.3%	170 2.1%	117 1.5%	90 1.1%	30 0.4%	35 0.4%	39.2% 60.8%
4	23 0.3%	45 0.6%	141 1.8%	395 4.9%	102 1.3%	122 1.5%	37 0.5%	41 0.5%	43.6% 56.4%
5	30 0.4%	56 0.7%	208 2.6%	142 1.8%	605 7.6%	93 1.2%	19 0.2%	44 0.5%	50.5% 49.5%
6	59 0.7%	30 0.4%	114 1.4%	145 1.8%	62 0.8%	521 6.5%	34 0.4%	52 0.7%	51.2% 48.8%
7	192 2.4%	136 1.7%	24 0.3%	30 0.4%	8 0.1%	18 0.2%	458 5.7%	119 1.5%	46.5% 53.5%
8	71 0.9%	204 2.5%	39 0.5%	41 0.5%	35 0.4%	73 0.9%	115 1.4%	435 5.4%	42.9% 57.1%
	49.0% 51.0%	46.4% 53.6%	34.8% 65.2%	39.5% 60.5%	60.5% 39.5%	52.1% 47.9%	45.8% 54.2%	43.5% 56.5%	46.5% 53.5%

Output Class	1	493 6.2%	36 0.4%	123 1.5%	36 0.4%	16 0.2%	49 0.6%	190 2.4%	54 0.7%	49.4% 50.6%
	2	33 0.4%	446 5.6%	38 0.5%	26 0.3%	48 0.6%	28 0.4%	108 1.4%	230 2.9%	46.6% 53.4%
	3	35 0.4%	15 0.2%	219 2.7%	123 1.5%	88 1.1%	64 0.8%	15 0.2%	28 0.4%	37.3% 62.7%
	4	39 0.5%	34 0.4%	205 2.6%	446 5.6%	91 1.1%	167 2.1%	55 0.7%	48 0.6%	41.1% 58.9%
	5	34 0.4%	55 0.7%	203 2.5%	139 1.7%	576 7.2%	101 1.3%	19 0.2%	59 0.7%	48.6% 51.4%
	6	63 0.8%	25 0.3%	139 1.7%	172 2.1%	114 1.4%	480 6.0%	11 0.1%	51 0.6%	45.5% 54.5%
	7	197 2.5%	141 1.8%	28 0.4%	23 0.3%	7 0.1%	18 0.2%	490 6.1%	139 1.7%	47.0% 53.0%
	8	106 1.3%	248 3.1%	45 0.6%	35 0.4%	60 0.8%	93 1.2%	112 1.4%	391 4.9%	35.9% 64.1%
		49.3% 50.7%	44.6% 55.4%	21.9% 78.1%	44.6% 55.4%	57.6% 42.4%	48.0% 52.0%	49.0% 51.0%	39.1% 60.9%	44.3% 55.7%
		1	2	3	4	5	6	7	8	

En este caso obtenemos una media de aciertos del **49.66%** y con un tiempo medio de entrenamiento de **180 segundos** aproximadamente (3 minutos).

***InitialLearnRate 0.01 y LearnRateDropFactor 0.11***

Se muestran a continuación las matrices de confusión obtenidas:

Output Class	1	479 6.0%	41 0.5%	75 0.9%	24 0.3%	6 0.1%	35 0.4%	157 2.0%	44 0.5%	55.6% 44.4%
	2	82 1.0%	592 7.4%	40 0.5%	16 0.2%	36 0.4%	26 0.3%	105 1.3%	183 2.3%	54.8% 45.2%
	3	74 0.9%	20 0.2%	245 3.1%	111 1.4%	64 0.8%	58 0.7%	11 0.1%	14 0.2%	41.0% 59.0%
	4	23 0.3%	11 0.1%	190 2.4%	510 6.4%	92 1.1%	151 1.9%	18 0.2%	19 0.2%	50.3% 49.7%
	5	33 0.4%	44 0.5%	274 3.4%	141 1.8%	690 8.6%	107 1.3%	19 0.2%	71 0.9%	50.0% 50.0%
	6	45 0.6%	26 0.3%	119 1.5%	155 1.9%	67 0.8%	553 6.9%	19 0.2%	45 0.5%	53.7% 46.3%
	7	207 2.6%	104 1.3%	24 0.3%	16 0.2%	7 0.1%	13 0.2%	556 7.0%	116 1.5%	53.3% 46.7%
	8	57 0.7%	162 2.0%	33 0.4%	27 0.3%	38 0.5%	57 0.7%	115 1.4%	508 6.3%	51.0% 49.0%
		47.9% 52.1%	59.2% 40.8%	24.5% 75.5%	51.0% 49.0%	69.0% 31.0%	55.3% 44.7%	55.6% 44.4%	50.8% 49.2%	51.7% 48.3%
		1	2	3	4	5	6	7	8	
	Target Class									

	1	2	3	4	5	6	7	8	
1	470 5.9%	41 0.5%	125 1.6%	32 0.4%	14 0.2%	45 0.6%	157 2.0%	46 0.6%	50.5% 49.5%
2	67 0.8%	584 7.3%	35 0.4%	31 0.4%	25 0.3%	14 0.2%	96 1.2%	217 2.7%	54.6% 45.4%
3	53 0.7%	11 0.1%	347 4.3%	120 1.5%	66 0.8%	68 0.9%	12 0.1%	15 0.2%	50.1% 49.9%
4	13 0.2%	28 0.4%	151 1.9%	492 6.2%	88 1.1%	141 1.8%	31 0.4%	26 0.3%	50.7% 49.3%
5	26 0.3%	35 0.4%	144 1.8%	112 1.4%	670 8.4%	58 0.7%	8 0.1%	51 0.6%	60.7% 39.3%
6	38 0.5%	16 0.2%	117 1.5%	149 1.9%	65 0.8%	600 7.5%	15 0.2%	38 0.5%	57.8% 42.2%
7	257 3.2%	89 1.1%	27 0.3%	24 0.3%	8 0.1%	18 0.2%	586 7.3%	94 1.2%	53.1% 46.9%
8	76 0.9%	196 2.5%	54 0.7%	40 0.5%	64 0.8%	56 0.7%	95 1.2%	513 6.4%	46.9% 53.1%
	47.0% 53.0%	58.4% 41.6%	34.7% 65.3%	49.2% 50.8%	67.0% 33.0%	60.0% 40.0%	58.6% 41.4%	51.3% 48.7%	53.3% 46.7%
	1	2	3	4	5	6	7	8	

	0	1	2	3	4	5	6	7	8
0	521 6.5%	45 0.6%	87 1.1%	34 0.4%	11 0.1%	34 0.4%	163 2.0%	36 0.4%	56.0% 44.0%
1	66 0.8%	543 6.8%	36 0.4%	26 0.3%	28 0.4%	25 0.3%	90 1.1%	201 2.5%	53.5% 46.5%
2	61 0.8%	15 0.2%	368 4.6%	145 1.8%	91 1.1%	75 0.9%	17 0.2%	19 0.2%	46.5% 53.5%
3	31 0.4%	24 0.3%	160 2.0%	463 5.8%	80 1.0%	126 1.6%	28 0.4%	25 0.3%	49.4% 50.6%
4	29 0.4%	39 0.5%	175 2.2%	133 1.7%	669 8.4%	63 0.8%	13 0.2%	50 0.6%	57.1% 42.9%
5	40 0.5%	20 0.2%	108 1.4%	138 1.7%	71 0.9%	593 7.4%	11 0.1%	49 0.6%	57.6% 42.4%
6	198 2.5%	93 1.2%	33 0.4%	27 0.3%	10 0.1%	15 0.2%	606 7.6%	89 1.1%	56.6% 43.4%
7	54 0.7%	221 2.8%	33 0.4%	34 0.4%	40 0.5%	69 0.9%	72 0.9%	531 6.6%	50.4% 49.6%
8	52.1% 47.9%	54.3% 45.7%	36.8% 63.2%	46.3% 53.7%	66.9% 33.1%	59.3% 40.7%	60.6% 39.4%	53.1% 46.9%	53.7% 46.3%

	1	2	3	4	5	6	7	8	
1	535 6.7%	41 0.5%	119 1.5%	38 0.5%	20 0.2%	62 0.8%	212 2.6%	53 0.7%	49.5% 50.5%
2	45 0.6%	479 6.0%	32 0.4%	38 0.5%	40 0.5%	25 0.3%	99 1.2%	236 2.9%	48.2% 51.8%
3	69 0.9%	18 0.2%	319 4.0%	162 2.0%	119 1.5%	87 1.1%	19 0.2%	20 0.2%	39.2% 60.8%
4	17 0.2%	27 0.3%	144 1.8%	419 5.2%	84 1.1%	133 1.7%	33 0.4%	29 0.4%	47.3% 52.7%
5	30 0.4%	62 0.8%	186 2.3%	133 1.7%	588 7.3%	91 1.1%	19 0.2%	61 0.8%	50.3% 49.7%
6	51 0.6%	36 0.4%	118 1.5%	154 1.9%	72 0.9%	530 6.6%	21 0.3%	54 0.7%	51.2% 48.8%
7	195 2.4%	123 1.5%	45 0.6%	32 0.4%	15 0.2%	21 0.3%	480 6.0%	132 1.7%	46.0% 54.0%
8	58 0.7%	214 2.7%	37 0.5%	24 0.3%	62 0.8%	51 0.6%	117 1.5%	415 5.2%	42.4% 57.6%
	53.5% 46.5%	47.9% 52.1%	31.9% 68.1%	41.9% 58.1%	58.8% 41.2%	53.0% 47.0%	48.0% 52.0%	41.5% 58.5%	47.1% 52.9%
	1	2	3	4	5	6	7	8	

	1	2	3	5	6	7	8		
1	463 5.8%	36 0.4%	73 0.9%	24 0.3%	9 0.1%	23 0.3%	169 2.1%	34 0.4%	55.7% 44.3%
2	83 1.0%	491 6.1%	60 0.8%	43 0.5%	42 0.5%	52 0.7%	118 1.5%	176 2.2%	46.1% 53.9%
3	52 0.7%	26 0.3%	173 2.2%	85 1.1%	47 0.6%	72 0.9%	50 0.6%	16 0.2%	33.2% 66.8%
4	37 0.5%	33 0.4%	230 2.9%	527 6.6%	195 2.4%	216 2.7%	54 0.7%	34 0.4%	39.7% 60.3%
5	52 0.7%	94 1.2%	329 4.1%	194 2.4%	634 7.9%	161 2.0%	20 0.2%	86 1.1%	40.4% 59.6%
6	52 0.7%	44 0.5%	61 0.8%	63 0.8%	27 0.3%	351 4.4%	35 0.4%	65 0.8%	50.3% 49.7%
7	189 2.4%	94 1.2%	33 0.4%	26 0.3%	8 0.1%	27 0.3%	424 5.3%	128 1.6%	45.6% 54.4%
8	72 0.9%	182 2.3%	41 0.5%	38 0.5%	38 0.5%	98 1.2%	130 1.6%	461 5.8%	43.5% 56.5%
	46.3% 53.7%	49.1% 50.9%	17.3% 82.7%	52.7% 47.3%	63.4% 36.6%	35.1% 64.9%	42.4% 57.6%	46.1% 53.9%	44.0% 56.0%
	1	2	3	5	6	7	8		

En este caso obtenemos una media de aciertos del **49.96%** y con un tiempo medio de entrenamiento de **169 segundos** aproximadamente (2 minutos 49 segundos).



**InitialLearnRate 0.011 y LearnRateDropFactor 0.1**

Se muestran a continuación las matrices de confusión obtenidas:

**Confusion Matrix**

Output Class \ Target Class	1	2	3	4	5	6	7	8	
1	514 6.4%	79 1.0%	183 2.3%	174 2.2%	68 0.8%	201 2.5%	266 3.3%	66 0.8%	33.1% 66.9%
2	93 1.2%	353 4.4%	95 1.2%	122 1.5%	98 1.2%	113 1.4%	131 1.6%	190 2.4%	29.5% 70.5%
3	33 0.4%	22 0.3%	183 2.3%	119 1.5%	135 1.7%	73 0.9%	14 0.2%	7 0.1%	31.2% 68.2%
4	32 0.4%	27 0.3%	122 1.5%	193 2.4%	104 1.3%	89 1.1%	48 0.6%	29 0.4%	30.0% 70.0%
5	69 0.9%	129 1.6%	249 3.1%	176 2.2%	445 5.6%	124 1.5%	43 0.5%	69 0.9%	34.1% 65.9%
6	34 0.4%	21 0.3%	52 0.6%	87 1.1%	40 0.5%	143 1.8%	43 0.5%	29 0.4%	31.8% 68.2%
7	159 2.0%	117 1.5%	56 0.7%	49 0.6%	43 0.5%	89 1.1%	276 3.4%	163 2.0%	29.0% 71.0%
8	66 0.8%	252 3.1%	60 0.7%	80 1.0%	67 0.8%	168 2.1%	180 2.2%	447 5.6%	33.9% 66.1%
	51.4% 48.6%	35.3% 64.7%	18.3% 81.7%	19.3% 80.7%	44.5% 55.5%	14.3% 85.7%	27.6% 72.4%	44.7% 55.3%	31.9% 68.1%

**Confusion Matrix**

Output Class \ Target Class	1	2	3	4	5	6	7	8	
1	510 6.4%	39 0.5%	124 1.5%	38 0.5%	10 0.1%	62 0.8%	226 2.8%	58 0.7%	47.8% 52.2%
2	33 0.4%	422 5.3%	39 0.5%	36 0.4%	32 0.4%	26 0.3%	91 1.1%	182 2.3%	49.0% 51.0%
3	46 0.6%	20 0.2%	262 3.3%	174 2.2%	102 1.3%	77 1.0%	17 0.2%	25 0.3%	36.2% 63.8%
4	27 0.3%	40 0.5%	164 2.0%	413 5.2%	76 0.9%	149 1.9%	49 0.6%	46 0.6%	42.8% 57.2%
5	32 0.4%	65 0.8%	202 2.5%	126 1.6%	620 7.7%	77 1.0%	25 0.3%	60 0.7%	51.4% 48.6%
6	58 0.7%	36 0.4%	132 1.6%	148 1.8%	79 1.0%	511 6.4%	27 0.3%	62 0.8%	48.5% 51.5%
7	202 2.5%	135 1.7%	31 0.4%	29 0.4%	12 0.1%	23 0.3%	439 5.5%	134 1.7%	43.7% 56.3%
8	92 1.1%	243 3.0%	46 0.6%	36 0.4%	69 0.9%	75 0.9%	127 1.6%	433 5.4%	38.6% 61.4%
	51.0% 49.0%	42.2% 57.8%	26.2% 73.8%	41.3% 58.7%	62.0% 38.0%	51.1% 48.9%	43.9% 56.1%	43.3% 56.7%	45.1% 54.9%

**Confusion Matrix**

Output Class \ Target Class	1	2	3	4	5	6	7	8	
1	498 6.2%	35 0.4%	131 1.6%	61 0.8%	35 0.4%	69 0.9%	255 3.2%	48 0.6%	44.0% 56.0%
2	52 0.6%	407 5.1%	50 0.6%	27 0.3%	55 0.7%	49 0.6%	126 1.6%	196 2.4%	42.3% 57.7%
3	59 0.7%	11 0.1%	164 2.0%	99 1.2%	63 0.8%	56 0.7%	21 0.3%	12 0.1%	33.8% 66.2%
4	25 0.3%	19 0.2%	141 1.8%	359 4.5%	142 1.8%	95 1.2%	51 0.6%	18 0.2%	42.2% 57.8%
5	42 0.5%	83 1.0%	280 3.5%	214 2.7%	476 5.9%	125 1.6%	16 0.2%	73 0.9%	36.4% 63.6%
6	65 0.8%	38 0.5%	159 2.0%	173 2.2%	132 1.6%	514 6.4%	44 0.5%	74 0.9%	42.9% 57.1%
7	175 2.2%	129 1.6%	32 0.4%	23 0.3%	11 0.1%	19 0.2%	370 4.6%	132 1.6%	41.5% 58.5%
8	84 1.0%	278 3.5%	43 0.5%	44 0.5%	86 1.1%	73 0.9%	118 1.5%	447 5.6%	38.1% 61.9%
	49.8% 50.2%	40.7% 59.3%	16.4% 83.6%	35.9% 64.1%	47.6% 52.4%	51.4% 48.6%	37.0% 63.0%	44.7% 55.3%	40.4% 59.6%

**Confusion Matrix**

Output Class \ Target Class	1	2	3	4	5	6	7	8	
1	7 0.1%	4 0.0%	6 0.1%	5 0.1%	5 0.1%	2 0.0%	7 0.1%	10 0.1%	15.2% 84.8%
2	5 0.1%	4 0.0%	8 0.1%	2 0.0%	2 0.0%	5 0.1%	8 0.1%	8 0.1%	9.5% 90.5%
3	31 0.4%	10 0.1%	61 0.8%	46 0.6%	49 0.6%	29 0.4%	20 0.2%	30 0.4%	22.1% 77.9%
4	20 0.2%	81 1.0%	130 1.6%	234 2.9%	71 0.9%	88 1.1%	25 0.3%	66 0.8%	32.7% 67.3%
5	109 1.4%	391 4.9%	439 5.5%	478 6.0%	620 7.7%	433 5.4%	139 1.7%	199 2.5%	22.1% 77.9%
6	178 2.2%	166 2.1%	140 1.7%	119 1.5%	147 1.8%	242 3.0%	137 1.7%	204 2.5%	18.2% 81.8%
7	484 6.0%	102 1.3%	134 1.7%	59 0.7%	35 0.4%	54 0.7%	381 4.8%	115 1.4%	27.9% 72.1%
8	166 2.1%	242 3.0%	82 1.0%	57 0.7%	71 0.9%	147 1.8%	284 3.5%	368 4.6%	26.0% 74.0%
	0.7% 99.3%	0.4% 99.6%	6.1% 93.9%	23.4% 76.6%	62.0% 38.0%	24.2% 75.8%	38.1% 61.9%	36.8% 63.2%	24.0% 76.0%

Confusion Matrix									
Output Class	1	2	3	4	5	6	7	8	
	540 6.7%	27 0.3%	119 1.5%	28 0.3%	15 0.2%	47 0.6%	182 2.3%	43 0.5%	53.9% 46.1%
	62 0.8%	533 6.7%	31 0.4%	16 0.2%	25 0.3%	23 0.3%	127 1.6%	214 2.7%	51.7% 48.3%
	53 0.7%	20 0.2%	293 3.7%	160 2.0%	93 1.2%	72 0.9%	24 0.3%	22 0.3%	39.8% 60.2%
	21 0.3%	20 0.2%	181 2.3%	475 5.9%	100 1.2%	131 1.6%	42 0.5%	32 0.4%	47.4% 52.6%
	31 0.4%	46 0.6%	204 2.5%	125 1.6%	651 8.1%	81 1.0%	15 0.2%	40 0.5%	54.6% 45.4%
	47 0.6%	27 0.3%	109 1.4%	132 1.6%	59 0.7%	553 6.9%	20 0.2%	45 0.6%	55.7% 44.3%
	171 2.1%	117 1.5%	22 0.3%	27 0.3%	8 0.1%	12 0.1%	471 5.9%	103 1.3%	50.6% 49.4%
	75 0.9%	210 2.6%	41 0.5%	37 0.5%	49 0.6%	81 1.0%	120 1.5%	501 6.3%	45.0% 55.0%
Target Class									
									54.0% 46.0%
									53.3% 46.7%
									29.3% 70.7%
									47.5% 52.5%
									65.1% 34.9%
									55.3% 44.7%
									47.1% 52.9%
									50.1% 49.9%
									50.2% 49.8%

En este caso obtenemos una media de aciertos del **38.32%** y con un tiempo medio de entrenamiento de **163 segundos** aproximadamente (2 minutos 43 segundos).

**InitialLearnRate 0.011 y LearnRateDropFactor 0.11**

Se muestran a continuación las matrices de confusión obtenidas:

Confusion Matrix									
Output Class	1	2	3	4	5	6	7	8	
	385 4.8%	124 1.5%	106 1.3%	47 0.6%	20 0.2%	94 1.2%	162 2.0%	97 1.2%	37.2% 62.8%
	137 1.7%	312 3.9%	68 0.8%	46 0.6%	24 0.3%	58 0.7%	131 1.6%	158 2.0%	33.4% 66.6%
	48 0.6%	24 0.3%	101 1.3%	76 0.9%	45 0.6%	53 0.7%	20 0.2%	14 0.2%	26.5% 73.5%
	30 0.4%	46 0.6%	106 1.3%	212 2.6%	120 1.5%	106 1.3%	34 0.4%	50 0.6%	30.1% 69.9%
	55 0.7%	79 1.0%	421 5.3%	363 4.5%	632 7.9%	217 2.7%	21 0.3%	56 0.7%	34.3% 65.7%
	79 1.0%	71 0.9%	94 1.2%	113 1.4%	92 1.1%	307 3.8%	41 0.5%	132 1.6%	33.0% 67.0%
	182 2.3%	159 2.0%	43 0.5%	72 0.9%	20 0.2%	36 0.4%	494 6.2%	138 1.7%	43.2% 56.8%
	84 1.0%	185 2.3%	61 0.8%	71 0.9%	47 0.6%	129 1.6%	98 1.2%	355 4.4%	34.3% 65.5%
Target Class									
									38.5% 61.5%
									31.2% 68.8%
									10.1% 89.9%
									21.2% 78.8%
									63.2% 36.8%
									30.7% 69.3%
									50.6% 49.4%
									35.5% 64.5%
									35.0% 65.0%

Confusion Matrix									
Output Class	1	2	3	4	5	6	7	8	
	383 4.8%	42 0.5%	129 1.6%	41 0.5%	19 0.2%	34 0.4%	173 2.2%	40 0.5%	44.5% 55.5%
	86 1.1%	352 4.4%	90 1.1%	93 1.2%	51 0.6%	67 0.8%	94 1.2%	179 2.2%	34.8% 65.2%
	22 0.3%	15 0.2%	30 0.4%	20 0.2%	11 0.1%	13 0.2%	16 0.2%	19 0.2%	20.5% 79.5%
	57 0.7%	80 1.0%	118 1.5%	221 2.8%	123 1.5%	112 1.4%	47 0.6%	38 0.5%	27.8% 72.2%
	46 0.6%	121 1.5%	461 5.8%	439 5.5%	666 8.3%	284 3.5%	28 0.3%	69 0.9%	31.5% 68.5%
	15 0.2%	24 0.3%	34 0.4%	54 0.7%	40 0.5%	79 1.0%	13 0.2%	40 0.5%	26.4% 73.6%
	278 3.5%	211 2.6%	74 0.9%	67 0.8%	39 0.5%	80 1.0%	500 6.2%	229 2.9%	33.8% 66.2%
	113 1.4%	155 1.9%	64 0.8%	65 0.8%	51 0.6%	331 4.1%	130 1.6%	386 4.8%	29.8% 70.2%
Target Class									
									38.3% 61.7%
									35.2% 64.8%
									3.0% 97.0%
									22.1% 77.9%
									66.6% 33.4%
									7.9% 92.1%
									50.0% 50.0%
									38.6% 61.4%
									32.7% 67.3%

Confusion Matrix									
Output Class	1	2	3	4	5	6	7	8	
	1 0.0%	4 0.0%	2 0.0%	5 0.1%	3 0.0%	2 0.0%	3 0.0%	4 0.0%	4.2% 95.8%
	1 0.0%	3 0.0%	4 0.0%	4 0.0%	2 0.0%	6 0.1%	7 0.1%	4 0.0%	9.7% 90.3%
	5 0.1%	22 0.3%	9 0.1%	10 0.1%	5 0.1%	24 0.3%	6 0.1%	6 0.1%	10.3% 89.7%
	243 3.0%	539 6.7%	541 6.8%	648 8.1%	716 8.9%	533 6.7%	271 3.4%	297 3.7%	17.1% 82.9%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	21 0.3%	16 0.2%	22 0.3%	24 0.3%	17 0.2%	14 0.2%	11 0.1%	17 0.2%	9.9% 90.1%
	729 9.1%	416 5.2%	422 5.3%	309 3.9%	257 3.2%	421 5.3%	703 8.8%	672 8.4%	17.9% 82.1%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
Target Class									
									0.1% 99.9%
									0.3% 99.7%
									0.9% 99.1%
									64.8% 35.2%
									0.0% 100%
									1.4% 98.6%
									70.2% 29.8%
									0.0% 100%
									17.2% 82.8%

Como se puede observar, la media de aciertos es de del **28.3%**, un valor bastante malo comparado con el resto de combinaciones. En cuanto a los tiempos de entrenamiento, la media obtenida fue de unos **154,812 segundos** (2 minutos 35 segundos).

## Selección del mejor resultado

Observando los datos presentados en la sección Estudio sobre el ajuste de los parámetros, parece que la mejor solución es la de *InitialLearnRate* **0.01** y *LearnRateDropFactor* **0.11**. Parece claro que las opciones con el *InitialLearnRate* en 0.011 no consiguen redes con un porcentaje de acierto por encima del 49% (obtenido con el *InitialLearnRate* a 0.01 para ambos valores de *LearnRateDropFactor*). Otro factor importante es que durante las diferentes ejecuciones (de las 20 totales por combinación) se observó que se tendía bastante a desestabilizar la red (por ejemplo, en el caso de *InitialLearnRate* 0.011 y *LearnRateDropFactor* 0.11, solo se pudieron obtener 3 matrices de confusión correctas).

En el caso de 0.01, ambas combinaciones rozan el 50%<sup>2</sup>, y los tiempos son bastante parecidos. Aún así, según los datos obtenidos parece un poco más eficiente la red neuronal entrenada con *InitialLearnRate* 0.01 y *LearnRateDropFactor* 0.11. Otra de las razones por las que se escoge esta combinación es la relativa al comportamiento del *Learning Rate* en los diferentes epochs. Tanto en con el fijando el valor del *LearnRateDropFactor* 0.1 como a 0.11 los resultados son bastante parecidos (tiende a variar un poco más en las últimas iteraciones con 0.11, pero son cifras despreciables).

## Explotación de la red

Para la explotación de la red se han descargado un par de imágenes y se han escalado para tener una resolución de 32x32 píxeles. A continuación se muestran las imágenes escogidas para esta parte (se pueden observar en la carpeta de *imagenes\_pruebas*):



Se tratan de cuatro imágenes pertenecientes a la clase *dog*, otras dos a la clase *airplane* y 2 a la clase de *horse*.

La red neuronal para comprobar que todo funciona correctamente debería de tener la misma eficiencia o una muy parecida, a la media obtenida. Como para elegir una red neuronal, uno de los indicadores más importantes es la matriz de confusión, es necesario obtener una red neuronal que tenga una tasa de acierto cercana al 50% (media obtenida de las 5 ejecuciones). A continuación se muestra matriz de confusión:

---

2 Es importante comentar que ejecutando el script del campos virtual con las 10 clases nunca obtuve valores cercanos a los mostrados en las diapositivas (70% de acierto); siempre fueron valores inferiores al 60%.



Confusion Matrix									
Output Class	1	2	3	4	5	6	7	8	
	505 6.3%	51 0.6%	84 1.0%	30 0.4%	12 0.1%	29 0.4%	146 1.8%	45 0.6%	56.0% 44.0%
	63 0.8%	524 6.5%	36 0.4%	29 0.4%	44 0.5%	20 0.2%	118 1.5%	187 2.3%	51.3% 48.7%
	78 1.0%	17 0.2%	317 4.0%	127 1.6%	114 1.4%	66 0.8%	23 0.3%	20 0.2%	41.6% 58.4%
	28 0.3%	32 0.4%	170 2.1%	479 6.0%	98 1.2%	134 1.7%	41 0.5%	31 0.4%	47.3% 52.7%
	26 0.3%	46 0.6%	209 2.6%	137 1.7%	610 7.6%	80 1.0%	13 0.2%	53 0.7%	52.0% 48.0%
	50 0.6%	31 0.4%	113 1.4%	145 1.8%	56 0.7%	582 7.3%	25 0.3%	58 0.7%	54.9% 45.1%
	186 2.3%	102 1.3%	35 0.4%	24 0.3%	10 0.1%	20 0.2%	535 6.7%	103 1.3%	52.7% 47.3%
	64 0.8%	197 2.5%	36 0.4%	29 0.4%	56 0.7%	69 0.9%	100 1.2%	503 6.3%	47.7% 52.3%
									50.5% 49.5%
									52.4% 47.6%
									31.7% 68.3%
									47.9% 52.1%
									61.0% 39.0%
									58.2% 41.8%
									53.4% 46.6%
									50.3% 49.7%
									50.7% 49.3%
									Target Class
									1
									2
									3
									4
									5
									6
									7
									8

Se puede apreciar que para la clase *airplane* tiene una tasa de acierto del 56% (clase 1), para la clase *dog* del 47.3% (clase 4) y para *horse* del 54.9% (clase 6). Para poder explotar la red neuronal obtenida (*net*) se modifica la parte de *Explotación de la red* para adaptarla a nuestros datos como se muestra a continuación (fichero *cifar\_pruebas.m*):

```
%% Explotacion de la red
imdsExplotacion = imageDatastore(fullfile(pwd, 'cifar10/imagenes_pruebas'),...
    'IncludeSubfolders',true,'LabelSource','foldernames');
YExplotacion = classify(net, imdsExplotacion);

% obtenemos las clases asociadas por la red neuronal
labels = {'airplane' 'automobile' 'bird' 'dog' 'frog' 'horse' 'ship' 'truck'};
for i=1:8
    YExplotacionClasses(i,:) = (YExplotacion == labels{i});
end

% Indicamos las salidas esperadas
TExplotacionClasses = [...
    0 0 0 0 0 0 1 1;...
    0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0;...
    1 1 1 1 0 0 0 0;...
    0 0 0 0 0 0 0 0;...
    0 0 0 0 1 1 0 0;...
    0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0;
];

% Generacion de la matriz de confusion
matrizConfusion = figure;
plotconfusion(double(YExplotacionClasses),double(TExplotacionClasses))
```




A continuación se muestra la salida obtenida en Matlab:

Confusion Matrix									
Output Class	1	2	3	4	5	6	7	8	
	2 25.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	0 0.0%	0 0.0%	1 12.5%	1 12.5%	1 12.5%	1 12.5%	0 0.0%	0 0.0%	25.0% 75.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 25.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
Target Class									
	1	2	3	4	5	6	7	8	
	100% 0.0%	NaN% NaN%	0.0% 100%	100% 0.0%	0.0% 100%	66.7% 33.3%	NaN% NaN%	NaN% NaN%	62.5% 37.5%

Como se puede observar, dadas las 8 imágenes, se obtuvo que la red neuronal entrenada obtiene una tasa de acierto del **62.5%**:

- las dos imágenes de aviones han sido correctamente clasificadas,
- las dos imágenes de caballos han sido correctamente clasificadas, y
- de las 4 imágenes de perros solo 1 (la segunda *dog2.jpg*) ha sido clasificada de forma correcta.

Vamos a analizar las imágenes mal clasificadas e intentar entender el motivo de la mala clasificación:

	Lo clasifica como un pájaro. Seguramente por la forma de la imagen, es importante comentar que en la clase de <i>bird</i> hay diferentes pájaros pequeños, pero también imágenes de avestruces.
	Lo clasifica como una rana. No entiendo los motivos de este caso.
	Lo clasifica como un caballo. Seguramente debido al color, al tamaño y a la forma parecida de un caballo (al ser imágenes pequeñas muchas características se pueden perder).

## Conclusiones

Durante la realización de esta práctica tuve varios problemas relacionados con los resultados obtenidos (especialmente fijándome en los valores de las matrices de confusión obtenidas, como se había comentado vía email), por lo que fue necesario ejecutar más de 10 iteraciones para cada combinación de *InitialLearnRate* y *LearnRateDropFactor*. A mayores, como también se había comentado, durante la ejecución del comando *cifar10CNN* de la sesión práctica de Deep Learning con las 10 clases de imágenes, los resultados obtenidos en las matrices de confusión (se ejecutaron ambos scripts del campus virtual) eran bastante pobres con los presentados en el pdf (una tasa de acierto del 75%, cuando en mi caso muy pocas veces superó el 50%).