

# **IMPACTO DEL DESEQUILIBRIO DE CLASES EN EL ENTRENAMIENTO DE REDES NEURONALES CONVOLUCIONALES EN PROBLEMAS MULTI-CLASE**

*IMPACT OF CLASS IMBALANCE IN THE TRAINING OF CONVOLUTIONAL NEURAL NETWORKS FOR MULTI-CLASS PROBLEMS*

**Andrés Ferreira Ramírez**

Universidad Autónoma Metropolitana  
*fra@correo.azc.uam.mx*

**Eduardo Rodríguez Martínez**

Universidad Autónoma Metropolitana  
*erm@correo.azc.uam.mx*

## **Resumen**

El problema del desequilibrio de clases en el aprendizaje automático, se presenta cuando el conjunto de entrenamiento subyacente está compuesto por un número desigual de muestras para cada clase, lo que ocasiona que datos de algunas clases dominen claramente. Aparentemente, la mayoría de los modelos clasificadores aprenden a clasificar dichos conjuntos de datos; sin embargo, presentan un rendimiento de generalización deficiente debido a un fuerte sesgo hacia las clases mayoritarias. En este artículo, se presenta un estudio sistemático dirigido a comprender como afecta el problema del desequilibrio de clases al rendimiento de una red neuronal convolucional entrenada para una tarea de clasificación de imágenes, y se presenta una metodología para corregir el sobreentrenamiento e incrementar la generalización de la red.

**Palabras Claves:** Aumento de datos, desequilibrio de clases, red neuronal convolucional, sobreentrenamiento.

## **Abstract**

*The class imbalance problem in machine learning occurs when the underlying training set is composed of unequal number of samples for each class, which causes*

*data from some classes to clearly dominate. Apparently, most classifiers learn to classify such datasets, however, they show poor generalization performance due to a strong bias towards the majority classes. This article presents a systematic study aimed at understanding how the class imbalance problem affects the performance of a convolutional neural network which has been trained for an image classification task. Also, we present a methodology to correct the overtraining and increase the generalization performance of the network.*

**Keywords:** Data augmentation, class imbalance, convolutional neural network, overfit.

## **1. Introducción**

La clasificación de imágenes es el proceso de asignar a una imagen una o múltiples etiquetas que describan su contenido. Para realizar la clasificación, se debe de diseñar un modelo que sea capaz de aprender las etiquetas que tiene que asignar a una imagen determinada; la asignación se realiza mediante un proceso de aprendizaje que utiliza un conjunto de imágenes de entrenamiento previamente etiquetadas, el cual debe de ser lo suficientemente grande para garantizar un entrenamiento eficiente.

En las últimas décadas, la clasificación de imágenes ha sido una de las áreas más estudiadas en el campo del aprendizaje automático, debido principalmente a su aplicación en problemas reales. Se han investigado muchos enfoques para encontrar soluciones óptimas en los problemas de clasificación, sin embargo, las bases de datos con grandes cantidades de imágenes y el aumento en la capacidad de procesamiento de las GPUs han hecho de las redes neuronales convolucionales (CNNs, por sus siglas en inglés) la mejor elección, ya que superan en rendimiento a los algoritmos tradicionales.

Actualmente, existen una gran variedad de bases de datos de imágenes etiquetadas, disponibles al público, que son utilizadas para desarrollar y comparar modelos de clasificación con CNN. Algunos de estos conjuntos de referencia contienen la misma cantidad de imágenes por clase y se conocen como conjuntos de datos equilibrados; sin embargo, la mayoría de estos contienen un número

desigual de imágenes y se conocen como conjuntos de datos desequilibrados. Tratar con desequilibrio de clases es un gran desafío ya que en estos conjuntos, los ejemplos de unas clases superan significativamente a los ejemplos de otras clases; además las clases minoritarias son las de mayor interés –desde el punto de vista del aprendizaje- ya que representan casos raros o por que la adquisición de datos de estos ejemplos es costosa.

En [Hensman and Masko, 2015] y [Pulgar, 2017] se establece que el desequilibrio de clases puede tener un efecto perjudicial significativo en el entrenamiento de los clasificadores basados en CNN. Para aliviar dicho efecto, en los últimos tres años han surgido diversas propuestas que adaptan las técnicas tradicionales para combatir el desequilibrio en los datos en el marco del aprendizaje profundo. Dichas técnicas se dividen en tres grupos: (a) muestreo a nivel de datos, en donde se modifica el conjunto de entrenamiento para disminuir los ejemplos de la clase mayoritaria [Huang, 2016] [Song, 2017] [Dong, 2018], o aumentar los ejemplos de la clase minoritaria [Ando, 2017]; (b) modificación a nivel algorítmico, cuyo objetivo es modificar el algoritmo de aprendizaje profundo a nivel de función objetivo [Huang, 2016] [Sze-To, 2017] [Dong, 2017]; y (c) sensibles a los costos, cuyo objetivo es penalizar en mayor grado los errores cometidos en las clases minoritarias [Wang et al., 2016] [Khan, 2018]. El enfoque más sencillo en las soluciones de muestreo a nivel de datos centrados en la clase mayoritaria consiste en eliminar aleatoriamente ejemplos de esta, sin embargo, existen técnicas que seleccionan los ejemplos a ser eliminados de forma que se minimice la pérdida de información [Song, 2017]. En el caso del muestreo centrado en la clase minoritaria, el enfoque más sencillo radica en duplicar ejemplos aleatoriamente, sin embargo, existen soluciones más inteligentes que generan ejemplos sintéticos nuevos mediante la interpolación entre ejemplos positivos preexistentes que se encuentran muy juntos (SMOTE) [Chawla, 2002] y soluciones en donde se sintetizan ejemplos buscando no alterar la distribución de las clases [Ando, 2017].

No obstante que las técnicas de muestreo aleatorio son fáciles de implementar y disminuyen considerablemente el nivel de desequilibrio, estas pueden dar lugar a un sobreajuste del modelo de clasificación. En el caso de las técnicas inteligentes

de generación o eliminación de ejemplos, estas son generalmente complejas y computacionalmente caras [He, 2009].

En este trabajo, se presenta un estudio sistemático para analizar el impacto del desequilibrio de clases en el rendimiento de clasificación de una CNN. Para el análisis se generan bases de datos con diferentes relaciones de desequilibrio utilizando un método de sub-muestreo que elimina muestras aleatorias de las clases minoritarias. Además, se presenta una metodología consistente en la aplicación de una serie de técnicas y algoritmos del estado del arte en un orden propuesto para corregir el sobreentrenamiento causado por el muestreo aleatorio e incrementar la generalización de la red.

## **2. Métodos**

En esta sección se proporcionan los detalles del método implementado para el desequilibrio de clases, se describe el algoritmo de aprendizaje seleccionado para este estudio y sus parámetros de entrenamiento, se describe el fenómeno de sobreajuste y el proceso de regularización para evitarlo, finalmente se describen los experimentos realizados.

### **Desequilibrio de clases**

El desequilibrio de clases puede tomar muchas formas, en particular en el contexto de clasificación multi-clase, típico en CNN. En [Buda, 2017] se define el desequilibrio por pasos, en el cual el número de ejemplos es igual dentro de las clases mayoritarias y minoritarias, pero difiere entre las clases mayoritarias y minoritarias. Este tipo de desequilibrio se caracteriza por dos parámetros: la fracción de las clases minoritarias definidas en la ecuación 1 y la relación de desequilibrio ( $IR$ ) entre el número de ejemplos en las clases mayoritarias y el número de ejemplos en las clases minoritarias definidas en la ecuación 2.

$$\mu = \frac{\{i \in \{1, \dots, N\} : C_i \text{ es minoritario}\}}{N} \quad (1)$$

$$\rho = \frac{\max_i\{|C_i|\}}{\min_i\{|C_i|\}} \quad (2)$$

donde  $C_i$  es un conjunto de ejemplos en la clase  $i$  y  $N$  es el número total de clases. En este trabajo, se usa el conjunto de datos Caltec101 [Caltech101, 2018], que consta de 9,144 imágenes a color de alta resolución pertenecientes a 103 categorías. Cada categoría contiene de 31 a 800 imágenes y la mayoría de las categorías tienen alrededor de 50 imágenes. El tamaño de cada imagen es de aproximadamente 300 x 200 píxeles. Se seleccionaron las 5 clases más pequeñas (clases minoritarias) y las 5 clase más grandes (clases mayoritarias), para formar un subconjunto con 1973 imágenes. El subconjunto fue dividido en 4 conjuntos con desequilibrio de clases, utilizando el parámetro  $\mu = 0.5$ , y diferentes relaciones de  $\rho$  (1:1, 3:1, 7:1, 15:1) para implementar el desequilibrio por pasos. Para la relación 1:1 se estandariza el número de imágenes por clase, a la clase con el menor número de imágenes, para formar un conjunto de 320 imágenes; este es el conjunto base para la formación de las diferentes relaciones de desequilibrio. Para la relación 3:1, se eliminan 16 imágenes de cada una de las clases minoritarias y se agregan 16 imágenes a cada una de las clases mayoritarias. Para la relación 7:1, se eliminan y se agregan 8 de las 16 imágenes restantes a las clases minoritarias y mayoritarias respectivamente; para la relación 15:1 se siguió el mismo procedimiento, ver la tabla 1 para más detalles. La estandarización, eliminación y el agregado de imágenes se realizó de manera aleatoria.

Tabla 1 Resumen de los conjuntos con desequilibrio de imágenes.

Clase	Num. De imágenes x clase	Relación de desequilibrio			
		1:1	3:1	7:1	15:1
Aviones	800	32	48	56	60
Caras	435	32	48	56	60
Reloj	239	32	48	56	60
Leopardos	200	32	48	56	60
Bonsai	128	32	48	56	60
Lirio de agua	37	32	16	8	4
Fresa	35	32	16	8	4
Gato montés	34	32	16	8	4
Binocular	33	32	16	8	4
Metronomo	32	32	16	8	4
<b>Totales</b>	<b>1973</b>	<b>320</b>	<b>320</b>	<b>320</b>	<b>320</b>

## Algoritmo de aprendizaje

Para la clasificación se utilizó una CNN que tiene una profundidad de 6 capas, 4 capas de convolución y 2 capas completamente conectadas. Las capas de convolución manejan de 32 a 512 filtros cuyo tamaño varía de 7x7 a 3x3. Una unidad de ReLu es aplicada a la salida de cada capa de convolución y capa completamente conectada. Una capa de pooling máximo es aplicada a la salida de cada capa de ReLu en cada capa de convolución. La salida de la segunda capa completamente conectada es alimentada a un Softmax que produce una distribución sobre las etiquetas de las  $N$  clases, tabla 2.

Tabla 2 CNN propuesta.

Capa	Tipo	Dimensión			Kernel	Paso	Relleno
		Ancho	Alto	Prof			
0	Input	227	227	3	-	-	-
1	Convolution_1	111	111	32	7	2	-
	ReLu_1	111	111	32	-	-	-
	Pooling max_1	55	55	32	3	2	-
2	Convolution_2	55	55	64	5	1	2
	ReLu_2	55	55	64	-	-	-
	Pooling max_2	27	27	64	3	2	-
3	Convolution_3	27	27	128	3	1	2
	ReLu_3	27	27	128	-	-	-
	Pooling max_3	13	13	128	3	2	-
4	Convolution_4	13	13	128	3	1	1
	ReLu_4	13	13	128	-	-	-
	Pooling max_4	6	6	128	-	-	-
5	Fully Connected_1	1	1	512	-	-	-
	ReLu_5	1	1	512	-	-	-
6	Fully Connected_2	1	1	NC	-	-	-
	Softmax	1	1	NC	-	-	-
	Clasificación	1	1	NC	-	-	-

La CNN fue entrenada utilizando descenso de gradiente estocástico con un tamaño de lote de 10, momento de 0.9 y una disminución de pesos (factor de regularización) de 0.0005. Los pesos iniciales en cada una de las capas fueron inicializados con una distribución gaussiana con una media de 0 y una desviación estándar de 0.01. Los umbrales de activación en cada una de las capas fueron inicializados a cero. El número de épocas de entrenamiento se fijó a 50. Se inició con una tasa de

aprendizaje de 0.001 la cual se disminuyó en un factor de 10 después de cada 10 épocas. La CNN fue entrenada en una GPU NVIDIA GeForce GTX 980M con 1536 núcleos y 4.3 GB de memoria, utilizando el Toolbox de redes neuronales de MatLab 2018a.

### **Sobreajuste**

Cuando se entrena una CNN, hay un compromiso entre la optimización (exactitud) y la generalización de la red. La optimización se refiere al proceso de ajuste de la red para obtener el mejor rendimiento posible con los datos de entrenamiento, mientras que la generalización se refiere a qué tan bien funciona la red entrenada con datos que nunca ha visto. En aprendizaje automático, el sobreajuste es definido como el efecto de sobreentrenar un algoritmo de aprendizaje con los datos utilizados para el entrenamiento. Durante el sobreajuste, la optimización se incrementa mientras la generalización de la red empeora, lo que indica, que la red quedó ajustada a características muy específicas de los datos de entrenamiento.

El proceso para luchar contra el sobreajuste de una CNN se conoce como regularización e incluye técnicas como:

- **Reducir el tamaño de la red:** El número de capas y el número de unidades por capa determinan el número de parámetros que se puede aprender la CNN, lo que se conoce como la capacidad del modelo. Un modelo con más parámetros tiene más capacidad de memorización y puede aprender fácilmente un mapeo perfecto entre las muestras de entrenamiento y sus objetivos, pero sin ningún poder de generalización; sin embargo, un modelo con menos parámetros y menos capacidad de memorización puede no aprender el mapeo tan fácilmente. Para encontrar el tamaño de la red apropiado para los datos, se deben de evaluar diferentes arquitecturas, comenzando con pocas capas y parámetros, y aumentando el tamaño de las capas o agregando capas nuevas a la red hasta encontrar rendimientos decrecientes con respecto a la pérdida de validación. Aunque reducir el tamaño de la red es la forma más sencilla de evitar el sobreajuste, las redes

grandes tienen el potencial de ser más poderosas que las redes pequeñas, por lo que esta es una opción que se adopta muy pocas veces.

- **Regularización de pesos:** Cuando se tiene una arquitectura de red fija y un conjunto de datos de entrenamiento fijos, pueden existir múltiples conjuntos de valores de pesos (modelos múltiples) que podrían describir los datos. Un modelo con pesos pequeños es más simple que un modelo con pesos grandes y tiene menos posibilidades de sobreajuste, bajo esta suposición implícita, una forma común de mitigar el sobreajuste es poner restricciones a la complejidad de la red al obligar a sus pesos a tomar valores pequeños, lo que hace que la distribución de los valores de los pesos sea más regular. Esto se llama regularización de pesos, y se hace agregando a la función de pérdida de la red una penalización por tener pesos grandes durante el entrenamiento. Una de las técnicas de regularización de pesos más utilizada en el contexto de las redes neuronales es la disminución de pesos (weight decay) o regularización  $L_2$ , en donde la penalización agregada es proporcional al cuadrado del valor del coeficiente de pesos. Efectivamente, esta regularización fomenta distribuciones de pesos pequeños y dispersos sobre grandes valores concentrados en solo unas pocas neuronas.
- **Dropout:** Es una técnica en la cual las salidas de algunas neuronas, seleccionadas aleatoriamente, se fijan a cero para que no contribuyan en el paso hacia adelante y no participen en la propagación hacia atrás del algoritmo de entrenamiento; lo que reduce coadaptaciones complejas de las neuronas, generadas en el aprendizaje estándar de retropropagación en redes neuronales. Las neuronas seleccionadas junto con sus conexiones entrantes y salientes, son marginadas de manera temporal, lo que equivale a muestrear una red “reducida” de la red original; la cual está formada con las  $n$  neuronas que sobrevivieron al marginado o aislamiento. La red con  $n$  unidades puede verse como una colección de  $2^n$  posibles redes neuronales reducidas, que comparten pesos de modo que la cantidad total de parámetros es menor. De esta manera, cada vez que se presenta un ejemplo de entrenamiento, la red neuronal muestrea y entrena una arquitectura

diferente. Durante el entrenamiento de la red, cada neurona se activa con una probabilidad fija (por lo general, 0.5 o se establece con un conjunto de validación). Este muestreo aleatorio de una subred dentro de la red de escala completa introduce un efecto de conjunto durante la fase de prueba, donde se usa la red completa para realizar la predicción. La técnica de Dropout o marginado [Srivastava, 2014], es uno de los enfoques más populares para la regularización de redes neuronales.

- **Aumento de datos:** El entrenamiento de una CNN con una gran cantidad de datos, es la clave para incrementar la generalización de la red. Sin embargo, en la práctica, la cantidad de datos que se tienen es limitada, lo que puede impedir el buen desarrollo del modelo. Para evitar este problema, es necesario crear datos falsos y agregarlos al conjunto de entrenamiento, técnica conocida como aumento de datos. Los enfoques más simples para aumentar los datos incluyen agregar ruido y aplicar transformaciones aleatorias a los datos existentes. En clasificación de imágenes, se puede agregar ruido gaussiano y se pueden aplicar transformaciones (aleatorias) simples a las imágenes, que incluyen: voltear (horizontal y verticalmente), rotación, escalar (el tamaño de la imagen), traslación (a lo largo de la dirección X o Y), etc. El aumento de datos puede aumentar el tamaño del conjunto de entrenamiento 10 veces o más, haciendo que el modelo sea más robusto (y evitar el sobreajuste).

## **Experimentos**

Para evaluar el rendimiento de la CNN, la red se entrenó desde cero con cada una de las relaciones de desequilibrio. Para las pruebas, cada base de datos fue dividida en conjunto de entrenamiento y prueba. El conjunto de entrenamiento fue formado con el 70% de las imágenes de la base de datos y el 30% restante se utilizó para formar el conjunto de prueba.

Para probar la corrección del sobreajuste de la CNN se realizaron tres pruebas. En la primera, se utilizó la CNN sin ningún tipo de regularización para observar el sobreajuste. En la segunda, se agregó a la CNN regularización de pesos, Dropout

y aumento de datos para ver su efecto en la reducción o eliminación del sobreajuste. Finalmente a la red implementada en la prueba dos, se le agregó normalización en diferentes capas con el fin de incrementar la generalización de la red. En estas pruebas, los parámetros para aumento de datos fueron: rango de rotación de -20 a 20 grados, reflexión aleatoria en la dirección izquierda-derecha, rango de traslación horizontal y vertical de -3 y 3 píxeles respectivamente. La tasa de regularización de pesos se fijó en 0.0005.

### 3. Resultados

#### Desequilibrio de clases

En la figura 1 se muestran los resultados del proceso de desequilibrio por pasos implementado con la CNN propuesta, se muestra las gráficas de exactitud para las fases de entrenamiento y prueba. En entrenamiento, la CNN llega a una exactitud del 100%, en todas las relaciones, en aproximadamente 20 épocas; sin embargo, en la fase de prueba obtiene exactitudes del 69% al 85%, lo que evidencia la falta de generalización de la red.

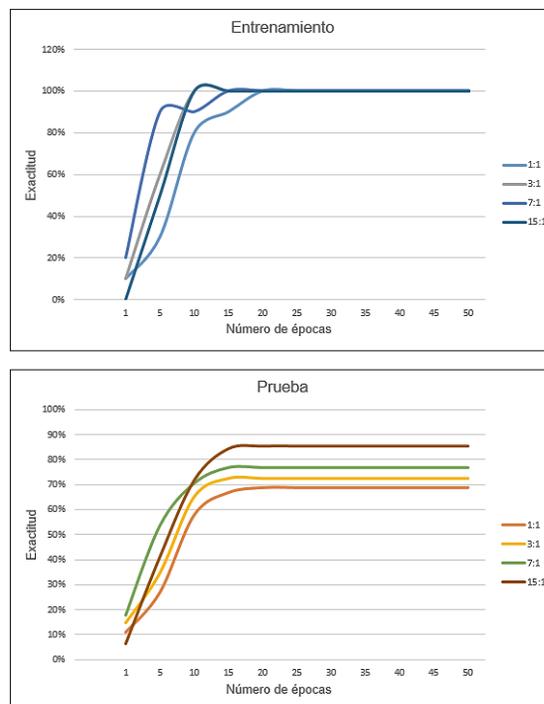


Figura 1 Gráficas de exactitud de la CNN entrenada en 50 épocas.

Note que, al incrementarse la relación de desequilibrio, la exactitud en prueba se incrementa; lo que podría suponer que la capacidad de generalización de la red mejora, pero no es así. En cada relación de desequilibrio, se incrementa el número de imágenes de las clases mayoritarias y se disminuye el de las clases minoritarias; esto causa que la red tienda a memorizar bien las clases mayoritarias y que vaya olvidando las minoritarias conforme la relación se hace más grande.

Aunque en la figura 1 se da información general sobre la exactitud de la CNN, en la tabla 3 se presentan los rendimientos de recuperación a nivel de clases (exactitud y recuerdo), con el fin de obtener información separada para las clases minoritarias (en color azul) y mayoritarias. Se define exactitud como la cantidad de muestras reconocidas correctamente en la clase  $i$  (auténticos positivos, AP) entre la cantidad de muestras reconocidas como de la clase  $i$  (AP más errores de tipo 1), mientras que recuerdo se define como el número de AP dividido entre el número de muestras que debieron ser reconocidas como de la clase  $i$  (AP más errores de tipo 2).

Tabla 3 Comparación de rendimiento de clases (CNN).

Clase	1:1		3:1		7:1		15:1	
	Exac	Rec	Exac	Rec	Exac	Rec	Exac	Rec
Aviones	0.875	0.700	0.867	0.929	0.889	0.941	1.000	0.944
Caras	1.000	0.700	0.688	0.786	0.923	0.706	0.857	1.000
Reloj	0.556	1.000	0.889	0.571	0.750	0.706	0.867	0.722
Leopardos	0.529	0.900	0.813	0.929	0.750	0.882	0.875	0.778
Bonsai	1.000	0.300	0.857	0.857	0.727	0.941	0.762	0.889
Lirio de agua	0.556	0.500	0.500	0.600	0.000	0.000	0.500	1.000
Fresa	0.900	0.900	0.429	0.600	0.333	0.500	0.000	0.000
Gato montés	0.429	0.600	0.286	0.400	0.000	0.000	0.000	0.000
Binocular	1.000	0.600	0.667	0.400	0.000	0.000	0.500	1.000
Metrónomo	0.875	0.700	1.000	0.400	0.500	0.500	1.000	1.000

Por ejemplo, para la clase “Caras” (relación 1:1), los resultados de la tabla 3 indican que ninguna de las muestras pertenecientes a las otras clases fue reconocida como “Caras”, mientras que el recuerdo indica que algunas muestras que pertenecen a la clase “Caras” fueron erróneamente asignadas a otras clases. Presentar dichas

métricas tiene por objetivo mostrar en que clases es en las que los errores de tipo 1 y tipo 2 predominan. Se puede observar que hay una tendencia clara de incremento y disminución en la exactitud para las clases mayoritarias y minoritarias respectivamente.

### **Sobreajuste**

En la figura 2 se muestran las gráficas de error de entrenamiento y prueba obtenidas para la CNN después de 50 épocas de entrenamiento. En algunos puntos de estas gráficas se puede observar que, mientras el error de entrenamiento disminuye constantemente, el error de prueba se incrementa; lo que demuestra que el desequilibrio de clases genera una situación de sobreajuste de la CNN. Para las relaciones 1:1 y 3:1 el sobreajuste se produce en las épocas 16 y 13 respectivamente; mientras que para las relaciones 7:1 y 15:1, la red sufre un sobreajuste parcial en la época 17.

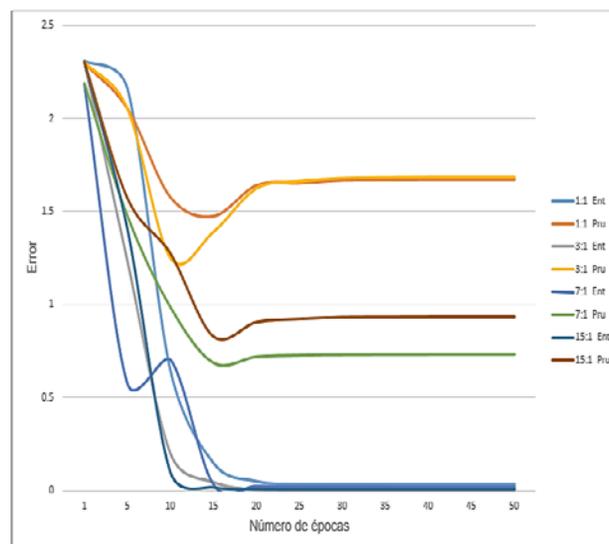


Figura 2 Error de entrenamiento (Ent) y prueba (Pru) del sobreajuste de la CNN.

En la figura 3, se muestran los resultados de las pruebas realizadas para corregir el sobreajuste de la red, se muestran las gráficas de exactitud y error para la fase de prueba. Se puede observar que la exactitud aumenta y el error disminuye cuando la CNN simple es regularizada con regularización de pesos, aumento de datos y

Dropout ( $RP + AD + Dropout$ ), y la exactitud mejora más cuando se agregan las capas de normalización a la CNN ( $RP + AD + Dropout + Normalización$ ).

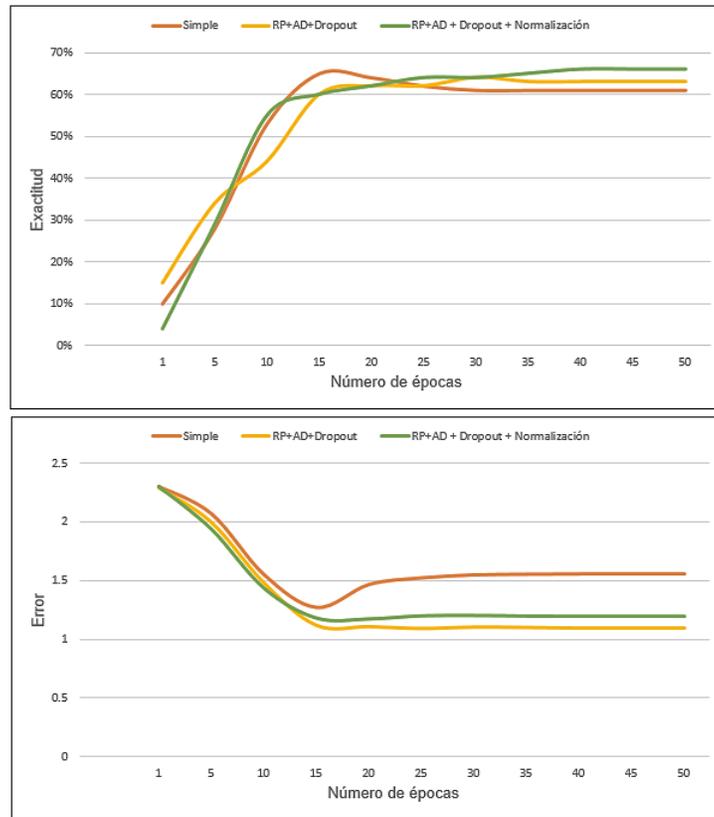


Figura 3 Resultados de corrección de sobreajuste de la CNN.

#### 4. Discusión

El desequilibrio de clases tiene un efecto perjudicial significativo en el entrenamiento de una CNN, afecta tanto la convergencia durante la fase de entrenamiento como la generalización de la red en el conjunto de prueba. La pérdida significativa de rendimiento se debe principalmente a la distribución asimétrica de clases, dada por la relación de desequilibrio. En este sentido, el uso de la medida de exactitud estándar, utilizada para guiar el proceso de aprendizaje de la CNN, puede estar proporcionando ventaja a la clasificación de las clases mayoritarias; además, las clases minoritarias, en la medida que se incrementa la relación de desequilibrio, pueden ser identificadas como ruido y, por lo tanto, pueden ser descartadas erróneamente por el clasificador.

El usar una relación de desequilibrio estática en la forma propuesta no equivale a transformar el problema de clasificación multi-clase en múltiples problemas de clasificación binaria, más bien, acentúa los casos de múltiples clases minoritarias y mayoritarias. Como se expone en [Wang, 2012], estos casos afectan negativamente el desempeño general del clasificador sobre las clases minoritarias. La razón por la que se utilizó una relación de desequilibrio estática es para trabajar en el peor de los casos que se pueden presentar en problemas desequilibrados multi-clase. Lo anterior nos indica que tenemos que ajustar adecuadamente la métrica, para medir el rendimiento del clasificador, a los objetivos perseguidos. Para aclarar esto, tomemos como ejemplo la relación 15:1 de nuestros experimentos, en donde las clases mayoritarias tienen 60 imágenes por clase y las minoritarias solo 4 imágenes. La clase aviones es el 18.75% del conjunto de 320 imágenes y la clase gato montés es el 1.25% del conjunto, pero estamos interesados en identificar gatos y no aviones. En este caso la red alcanzó una exactitud del 94.4% prediciendo la clase aviones y una exactitud de 0% prediciendo la clase gato montés, todo el tiempo; lo que genera un clasificador inútil para el objetivo planteado. En cambio, una métrica adecuadamente ajustada podría lograr una exactitud menor prediciendo aviones, pero tendría una tasa de recuperación sustancialmente mayor, que es realmente la medida para la que debería haberse optimizado.

## **5. Conclusiones**

Ya que muchas de las CNN empleadas para problemas de clasificación de imágenes utilizan conjuntos de entrenamiento desbalanceados, esto puede generar modelos de clasificación sub-óptimos; es decir, una buena cobertura de los ejemplos de las clases mayoritarias, mientras que los ejemplos de las clases minoritarias son con frecuencia clasificados erróneamente. Por lo tanto, esos algoritmos que obtienen un buen comportamiento en el marco de la clasificación con CNN, no necesariamente logran el mejor rendimiento para los conjuntos de datos desequilibrados.

El desequilibrio de clases provoca el sobreajuste de la CNN a los datos de entrenamiento, lo que afecta su capacidad de generalización. Las técnicas de

regularización de pesos, aumento de datos y dropout, funcionan correctamente para reducir el sobreentrenamiento de la CNN pero no ayudan a incrementar el rendimiento en prueba. Las capas de normalización integradas en la CNN, incrementan la exactitud en prueba en un 10%, es decir, mejoran la generalización de la CNN.

## **6. Bibliografía y Referencias**

- [1] Ando, S., Huang, C., Deeper over-sampling framework for classifying imbalance data. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD'17), 770-785, 2017.
- [2] Buda, M., Maki, A., Mazurowski, M., A systematic study of the class imbalance problem in convolutional neural networks. Computer Vision and Pattern Recognition, journal (CoRR), arXiv preprint arXiv:1710.05381, 2017.
- [3] Caltech101, Computational Vision at Caltech: [http://www.vision.caltech.edu/Image\\_Datasets/Caltech101/](http://www.vision.caltech.edu/Image_Datasets/Caltech101/).
- [4] Chawla, N., Bowyer, K., Kegelmeyer, W., SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, Vol. 16, 321-357, 2002.
- [5] Dong, Q., Gong, S., Zhu, X., Class rectification hard mining for imbalanced deep learning. In Proceedings of the IEEE International Conference on Computer Vision (ICCV'17), 1869-1878, 2017.
- [6] Dong, Q., Gong, S., Zhu, X., Imbalanced deep learning by minority class incremental rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence, Early Access, 2018.
- [7] He, H., Garcia, E., Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, Vol. 21, Issue 9, 1263-1284, 2009.
- [8] Hensman, P., Masko, D., The impact of imbalanced training data for convolutional neural networks. Degree project, 2015.
- [9] Khan, S., Hayat, M., Bennamoun, M., Sohel, F., Tongeri, R., Cost-sensitive learning of deep feature representations from imbalanced data. IEEE

- Transactions on Neural Networks and Learning Systems, Vol. 29, Issue 8, 3573-3587, 2018.
- [10] Huang, C., Li, Y., Loy, C., Tang, X., Learning deep representation for imbalanced classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16), 5375-5384, 2016.
- [11] Pulgar, F., Rivera, A., Charte, F., del Jesus, M., On the impact of imbalanced data in convolutional neural networks performance. In Proceedings of the International Conference on Hybrid Artificial Intelligence Systems (HAIS'17), 220-232, 2017.
- [12] Song, J., Shen, Y., Jing, Y., Song, M., Towards deeper insights into deep learning from imbalanced data. In Proceedings of the Chinese Conference on Computer Vision (CCCV'17), 674-684, 2017.
- [13] Srivastava, N., Hilton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, Vol. 15, 1929-1958, 2014.
- [14] Sze-To, A., Wong, A., A weight-selection strategy on training deep neural networks for imbalanced classification. In Proceedings of the International Conference on Image Analysis and Recognition (ICAR'2017), 3-10, 2017.
- [15] Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q., Kennedy, P., Training deep neural networks on imbalanced data sets. In Proceedings of the International Joint Conference on Neural Networks (IJCNN'16), 4368-4374, 2016.
- [16] Wang, S., Yao, X., Multiclass imbalance problems: Analysis and potential solutions. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, Vol. 42, Issue 4, 1119-1130, 2012.