

ANÁLISIS ESTADÍSTICO DE PESOS ENTRENADOS EN YOLOv1: CORRELACIONES ENTRE PROFUNDIDAD DE RED, ESPARSIDAD Y ENTROPÍA DE SHANNON EN CAPAS CONVOLUCIONALES

Eduardo Laura Pizarro ¹

1. Instituto de Electrónica Aplicada (IEA), Universidad Mayor de San Andrés, La Paz, Bolivia
elaura37@umsa.bo | ORCID: 0009-0004-1496-7893

RESUMEN

La arquitectura YOLOv1 revolucionó la detección de objetos en tiempo real al reformular la tarea como un problema de regresión de una sola etapa [1]. Sin embargo, su comportamiento interno permanece bajo el paradigma de "caja negra", existiendo una carencia de estudios que evalúen formalmente la distribución de sus tensores de pesos entrenados. El presente trabajo tiene como objetivo evaluar cuantitativamente la eficiencia y redundancia paramétrica en las 24 capas convolucionales de YOLOv1 mediante un diagnóstico estadístico estructural post-entrenamiento. Se aplicó un proceso de ingeniería inversa sobre el archivo binario yolov1.weights para extraer los tensores de cada capa, calculando cuatro métricas: Esparsidad (S), Entropía de Shannon (H), Norma de Frobenius ($\|W\|_F$) y Eficiencia Paramétrica (η_c). Los resultados revelan una correlación muy fuerte entre la profundidad de la red y la eficiencia paramétrica (Spearman $\rho = 0.965$, $p < 0.001$), así como una correlación negativa moderada entre esparsidad y entropía ($\rho = -0.691$, $p < 0.001$), indicando que las capas con mayor redundancia presentan menor diversidad informativa. Se identificó una jerarquía de importancia paramétrica donde las capas profundas concentran la mayor magnitud de pesos ($\|W\|_F = 16.391$) y eficiencia computacional ($\eta_c = 1.02 \times 10^7$). Este estudio proporciona un marco analítico-cuantitativo fundamentado en la teoría de la información y el álgebra lineal para identificar cuellos de botella estructurales y componentes idóneos para futuras fases de compresión y poda de modelos de visión artificial.

Palabras clave: YOLOv1, esparsidad, eficiencia paramétrica, detección de objetos.

1. INTRODUCCIÓN

La detección de objetos es una tarea de visión artificial que es un subcampo de la Inteligencia Artificial, que consiste en identificar y localizar objetos en imágenes o vídeos. Esta tarea no solo reconoce los objetos presentes en una imagen o vídeo, sino que también proporciona información sobre su ubicación mediante el dibujo de recuadros delimitadores a su alrededor. Antes de la introducción de YOLO v1, existían algoritmos de detección de objetos como R-CNN y Fast R-CNN, que seguían un enfoque de dos etapas [2,3]. Las principales desventajas de los algoritmos de detección de objetos en dos etapas, como R-CNN y Fast R-CNN, son su complejidad y su menor velocidad de procesamiento. Este proceso de dos pasos puede ser computacionalmente intensivo y lento, especialmente al aplicarse a grandes conjuntos de datos o escenarios en tiempo real. El 8 de junio de 2015 fue publicado por primera vez el artículo fundamental de YOLO versión 1, "You Only Look Once: Unified, Real-Time Object Detection v1" [1] en el repositorio de preimpresiones arXiv.

YOLO revolucionó el enfoque para la detección de objetos, que reformula la tarea como un problema de regresión de una sola etapa. En lugar de utilizar dos etapas separadas para la predicción de cuadros delimitadores y la predicción de categorías, como los métodos anteriores, YOLO emplea una única red para predecir simultáneamente los cuadros delimitadores y las categorías de objetos en un solo paso, lo que agiliza el proceso y logra una eficiencia en tiempo real.

La arquitectura original de YOLOv1 utiliza una única CNN que procesa la imagen completa de una sola vez. Esta red consta de 24 capas convolucionales seguidas de 2 capas (local y conectada), las otras capas no son "capas entrenables", porque no son componentes esenciales del pipeline.

Tabla 1
Capas YOLOV1

N	TIPO DE CAPA	FILTROS/ CANALES	TAMAÑO (SIZE)	STRIDE	PAD	BATCH NORM	ACTIVACIÓN	SALIDA/ PARÁMETROS
1	Convolutional	64	7×7	2	1	Sí	LeakyReLU	-
2	Maxpool	-	2×2	2	-	No	-	-
3	Convolutional	192	3×3	1	1	Sí	LeakyReLU	-
4	Maxpool	-	2×2	2	-	No	-	-
5	Convolutional	128	1×1	1	1	Sí	LeakyReLU	-

6	Convolutional	256	3×3	1	1	Sí	LeakyReLU	-
7	Convolutional	256	1×1	1	1	Sí	LeakyReLU	-
8	Convolutional	512	3×3	1	1	Sí	LeakyReLU	-
9	Maxpool	-	2×2	2	-	No	-	-
10	Convolutional	256	1×1	1	1	Sí	LeakyReLU	-
11	Convolutional	512	3×3	1	1	Sí	LeakyReLU	-
12	Convolutional	256	1×1	1	1	Sí	LeakyReLU	-
13	Convolutional	512	3×3	1	1	Sí	LeakyReLU	-
14	Convolutional	256	1×1	1	1	Sí	LeakyReLU	-
15	Convolutional	512	3×3	1	1	Sí	LeakyReLU	-
16	Convolutional	256	1×1	1	1	Sí	LeakyReLU	-
17	Convolutional	512	3×3	1	1	Sí	LeakyReLU	-
18	Convolutional	512	1×1	1	1	Sí	LeakyReLU	-
19	Convolutional	1024	3×3	1	1	Sí	LeakyReLU	-
20	Maxpool	-	2×2	2	-	No	-	-
21	Convolutional	512	1×1	1	1	Sí	LeakyReLU	-
22	Convolutional	1024	3×3	1	1	Sí	LeakyReLU	-
23	Convolutional	512	1×1	1	1	Sí	LeakyReLU	-
24	Convolutional	1024	3×3	1	1	Sí	LeakyReLU	-
25	Convolutional	1024	3×3	1	1	Sí	LeakyReLU	-
26	Convolutional	1024	3×3	2	1	Sí	LeakyReLU	-
27	Convolutional	1024	3×3	1	1	Sí	LeakyReLU	-
28	Convolutional	1024	3×3	1	1	Sí	LeakyReLU	-
29	Local	256	3×3	1	1	No	LeakyReLU	-
30	Dropout	-	-	-	-	No	-	prob=0.5
31	Connected	-	-	-	-	No	Linear	output=1715
32	Detection	-	-	-	-	No	-	classes=20, coords=4, side=7, num=3

Fuente: Elaboración propia.

A pesar del éxito de las redes neuronales convolucionales (CNN) en tareas de visión artificial en tiempo real, su comportamiento interno sigue estando dominado por el paradigma de la 'caja negra'. En la arquitectura fundacional YOLOv1, existe una profunda carencia de estudios que evalúen de manera estática y formal la distribución de sus tensores de pesos entrenados (w). Aunque el rendimiento global del modelo en términos de precisión (mAP) y costo computacional acumulado (BFLOPs) está ampliamente documentado, se desconoce cómo interactúan la redundancia paramétrica y la eficiencia de la información a nivel de capas individuales.

El problema científico radica en la ausencia de un marco analítico-cuantitativo que correlacione la profundidad de la red con el uso eficiente del espacio de parámetros. Actualmente, la literatura no ofrece una respuesta matemática clara a las siguientes ineficiencias estructurales de la arquitectura:

La Redundancia por Esparsidad Estática: No se ha determinado si el volumen de pesos cuya magnitud absoluta es inferior a un umbral crítico ($|w_i| < \epsilon$) crece de forma lineal o si se concentra en bloques específicos, lo que indicaría la existencia de subredes inactivas que consumen memoria innecesariamente.

La Homogeneidad Informativa (Entropía): Se ignora si la distribución de probabilidad $P(w_i)$ de los filtros convolucionales sufre de un colapso de entropía de Shannon (H). Una entropía anormalmente baja en capas profundas revelaría que el modelo concentra sus pesos en rangos extremadamente estrechos, perdiendo capacidad de representación y duplicando información semántica.

El Desacoplamiento Costo-Magnitud: No existe un diagnóstico que contraste la "energía" o importancia de los filtros medida a través de su Norma de Frobenius ($\|W\|_F$) frente a la densidad paramétrica real por unidad de cómputo (η_c). Esto impide identificar formalmente los cuellos de botella computacionales: capas que exigen un alto número de operaciones binarias (BFLOPs) pero cuyos filtros poseen normas despreciables, convirtiéndose en candidatas críticas para la poda (pruning) [6].

Por lo tanto, este trabajo resuelve la necesidad científica de formular un diagnóstico estadístico riguroso sobre el modelo entrenado de YOLOv1. Mediante la cuantificación explícita de estas variables, se busca transformar el proceso empírico de optimización de redes en un procedimiento analítico fundamentado en las leyes de la teoría de la información y el álgebra lineal.

El objetivo general de este trabajo es evaluar cuantitativamente la eficiencia y la redundancia paramétrica en las 24 capas convolucionales de la arquitectura YOLOv1, mediante el análisis estadístico de sus pesos entrenados, con el fin de identificar analíticamente los cuellos de botella de información y los componentes estructurales idóneos para futuras fases de compresión y optimización del modelo.

2. METODOLOGÍA

El presente artículo se desarrolló bajo un enfoque cuantitativo y de carácter experimental, orientado al diagnóstico estadístico estructural de una arquitectura de red neuronal convolucional (CNN) profunda post-entrenamiento.

2.1. MUESTRA Y ORIGEN DE LOS DATOS

La muestra de datos analizada en este estudio se obtuvo mediante un proceso de ingeniería inversa aplicada sobre el archivo binario yolov1.weights. Dado que los pesos entrenados de una red neuronal profunda son, por diseño, un bloque compacto y opaco, el origen de los datos no fue una fuente pública preprocesada, sino la reconstrucción de la arquitectura de pesos a partir de su representación binaria original.

Para acceder a esta información, se desarrolló un módulo extractor de pesos, el proceso de extracción se llevó a cabo en un entorno de sistema operativo Kali Linux (kernel Linux 6.18.12+kali-amd64), mediante un ejecutable personalizado en lenguaje C que interactúa con el framework Darknet. Este procedimiento consistió en la deserialización de los tensores de cada capa convolucional, logrando desarticular el archivo único de pesos en 24 matrices individuales correspondientes a las capas convolucionales de la red. Cada matriz, que contiene los parámetros (w) aprendidos durante el entrenamiento, fue extraída con su precisión original de punto flotante para asegurar la fidelidad del análisis estadístico.

Este enfoque de ingeniería inversa permitió transformar una estructura binaria estática en un conjunto de datos estructurados, exportados al formato .csv.

La tabla a continuación resume las 24 capas convolucionales extraídas, las cuales constituyen el corpus de trabajo para nuestro diagnóstico paramétrico:

Tabla 2
Capas Convolucionales extraídas de los pesos entrenados de YOLOv1

N°	CAPA	TIPO	ARCHIVO CSV EXPORTADO	RUTA COMPLETA
1	0	Conv	layer_0_weights.csv	analisis_pesos/layer_0_weights.csv
2	2	Conv	layer_2_weights.csv	analisis_pesos/layer_2_weights.csv
3	4	Conv	layer_4_weights.csv	analisis_pesos/layer_4_weights.csv
4	5	Conv	layer_5_weights.csv	analisis_pesos/layer_5_weights.csv
5	6	Conv	layer_6_weights.csv	analisis_pesos/layer_6_weights.csv
6	7	Conv	layer_7_weights.csv	analisis_pesos/layer_7_weights.csv
7	9	Conv	layer_9_weights.csv	analisis_pesos/layer_9_weights.csv
8	10	Conv	layer_10_weights.csv	analisis_pesos/layer_10_weights.csv
9	11	Conv	layer_11_weights.csv	analisis_pesos/layer_11_weights.csv
10	12	Conv	layer_12_weights.csv	analisis_pesos/layer_12_weights.csv
11	13	Conv	layer_13_weights.csv	analisis_pesos/layer_13_weights.csv
12	14	Conv	layer_14_weights.csv	analisis_pesos/layer_14_weights.csv
13	15	Conv	layer_15_weights.csv	analisis_pesos/layer_15_weights.csv
14	16	Conv	layer_16_weights.csv	analisis_pesos/layer_16_weights.csv
15	17	Conv	layer_17_weights.csv	analisis_pesos/layer_17_weights.csv
16	18	Conv	layer_18_weights.csv	analisis_pesos/layer_18_weights.csv
17	20	Conv	layer_20_weights.csv	analisis_pesos/layer_20_weights.csv
18	21	Conv	layer_21_weights.csv	analisis_pesos/layer_21_weights.csv
19	22	Conv	layer_22_weights.csv	analisis_pesos/layer_22_weights.csv
20	23	Conv	layer_23_weights.csv	analisis_pesos/layer_23_weights.csv
21	24	Conv	layer_24_weights.csv	analisis_pesos/layer_24_weights.csv
22	25	Conv	layer_25_weights.csv	analisis_pesos/layer_25_weights.csv
23	26	Conv	layer_26_weights.csv	analisis_pesos/layer_26_weights.csv
24	27	Conv	layer_27_weights.csv	analisis_pesos/layer_27_weights.csv

Fuente: Elaboración propia.

Los datos fuente consistieron en los tensores de pesos (w) extraídos directamente del archivo binario yolov1.weights. El proceso de extracción se llevó a cabo en un entorno de sistema operativo Kali Linux (kernel 6.18.12), mediante un ejecutable personalizado en lenguaje C que interactúa con el framework Darknet. Este procedimiento permitió interceptar los valores en punto flotante de cada filtro convolucional antes de cualquier etapa de procesamiento o inferencia.

2.2. DEFINICIÓN OPERACIONAL DE LAS VARIABLES

2.2.1. EVALUACIÓN DE LA REDUNDANCIA PARAMÉTRICA

La redundancia ocurre cuando muchos pesos "dicen lo mismo" o son irrelevantes.

Análisis de Esparsidad (Sparsity Ratio):

Matemáticamente, definimos una tolerancia ϵ (por ejemplo, $\epsilon=0.001$). La redundancia por esparsidad se calcula como:

$$S = \frac{1}{N} \sum_{i=1}^N [|w_i| < \epsilon] \quad (1)$$

Donde w_i es cada peso y $[\cdot]$ es el corchete de Iverson (que vale 1 si la condición es verdadera y 0 si no). Si S es muy alto, se tiene una alta redundancia.

Entropía de la Distribución:

La entropía de Shannon nos dice cuánta "información" aporta la distribución de pesos. Si la entropía es muy baja, significa que los pesos están altamente concentrados en un rango estrecho, lo que sugiere redundancia [4].

$$H(W) = - \sum_i p(w_i) \log p(w_i) \quad (2)$$

Donde $p(w_i)$ es la probabilidad de encontrar un peso en un rango específico.

2.2.2. EVALUACIÓN DE LA EFICIENCIA DEL MODELO

La eficiencia se mide como el rendimiento (precisión) en relación con el costo computacional (parámetros).

Densidad Paramétrica vs. Capas:

Debes calcular la "Eficiencia por Capa" (η_c). Si una capa tiene muchos parámetros, pero aporta poca varianza a la salida (calculada mediante el valor propio de las activaciones), esa capa es ineficiente.

$$\eta_c = \frac{\text{Número de Parámetros en Capa } c}{\text{BFLOPs de la Capa } c} \quad (3)$$

Norma de Frobenius de los Filtros:

Para ver qué capas son las más "activas" o "importantes" matemáticamente, calculamos la norma de las matrices de pesos [5]:

$$\|W\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |w_{ij}|^2} \quad (4)$$

Procedimiento experimental

Las capas con normas muy pequeñas son candidatas a ser eliminadas (podadas) sin afectar significativamente la salida de la red.

1. Extracción de Pesos mediante Ingeniería Inversa

Se desarrolló un módulo de extracción integrado en el framework Darknet, programado en lenguaje C. Este módulo realizó una lectura secuencial del archivo binario yolov1.weights para deserializar cada capa convolucional. Mediante la API interna del framework, se iteró a través de los tensores de pesos (W) de las 24 capas convolucionales, exportando cada conjunto de parámetros en formato de texto plano (.csv) con precisión de punto flotante. Este paso permitió convertir una estructura de datos propietaria y opaca en un repositorio tabular accesible para su manipulación estadística.

2. Preprocesamiento y Normalización

Previo al cálculo de las métricas de información, los tensores extraídos fueron sometidos a un proceso de limpieza y escalamiento. Para el cálculo específico de la Entropía de Shannon (H), los valores de cada capa se normalizaron mediante el escalamiento Min-Max, confinando el rango de los pesos al intervalo $[0,1]$ conforme a la Ecuación (5):

$$w'_i = \frac{w_i - \min(W)}{\max(W) - \min(W) + \xi} \quad (5)$$

Donde $\xi=1 \times 10^{-9}$ se utilizó como término de estabilización para evitar divisiones por cero. Esta normalización fue crítica para garantizar que la distribución de probabilidad $P(w_j)$ fuera comparable entre capas, independientemente de sus escalas originales de inicialización. Por el contrario, las métricas de Sparsity y Norma de Frobenius se calcularon directamente sobre los tensores originales W para preservar la integridad de la magnitud física de las conexiones neuronales.

3. Análisis Estadístico y Correlación

Tras el cálculo de las métricas $(S, H, \|W\|_F, \eta_c)$, se consolidó una matriz de datos maestra. Para identificar las dependencias estructurales de la arquitectura, se aplicó el Coeficiente de Correlación de Pearson (r) sobre el conjunto completo de variables. La correlación se calculó para evaluar la fuerza y dirección de las relaciones lineales entre la profundidad (Capa) y las métricas de desempeño, así como la interacción entre la redundancia y la riqueza informativa del modelo. Todo el procesamiento se realizó en un entorno Python 3.13, utilizando las librerías Pandas, NumPy y SciPy para asegurar la precisión numérica.

3. RESULTADOS

Se presenta el análisis cuantitativo de los parámetros aprendidos por la arquitectura YOLOv1. El objetivo es evaluar la eficiencia paramétrica, la distribución de la información y la redundancia interna de la red mediante métricas estadísticas aplicadas a los tensores de pesos de las 24 capas convolucionales principales.

Tabla 3
Capas Convolucionales principales de acuerdo a los pesos

Nº	CAPA	SPARSITY	ENTROPIA	NORMA_FROBENIUS	EFICIENCIA_ETA	LOG_EFICIENCIA_ETA
1	0	0.050064	2.292842	4.374817	9.966E+03	3.999
2	2	0.132134	1.495050	5.409610	3.985E+04	4.600
3	4	0.044312	2.620331	3.985896	1.596E+05	5.203

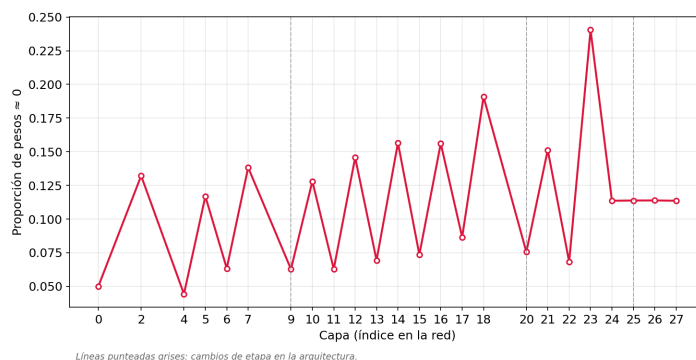
4	5	0.116899	1.622893	5.966264	1.594E+05	5.203
5	6	0.063339	1.969748	4.465167	1.595E+05	5.203
6	7	0.138267	1.825949	7.785106	1.594E+05	5.203
7	9	0.062782	2.338898	5.714068	6.363E+05	5.804
8	10	0.128020	1.787742	7.969863	6.376E+05	5.805
9	11	0.062851	2.167965	5.442551	6.363E+05	5.804
10	12	0.145677	1.797640	6.855215	6.376E+05	5.805
11	13	0.069412	2.040769	4.799645	6.363E+05	5.804
12	14	0.156510	1.779521	6.495820	6.376E+05	5.805
13	15	0.073555	2.180171	4.787890	6.363E+05	5.804
14	16	0.156154	1.693209	6.743656	6.376E+05	5.805
15	17	0.086533	1.939970	5.668434	6.378E+05	5.805
16	18	0.190764	1.480743	10.299357	6.377E+05	5.805
17	20	0.075531	1.729623	8.610216	2.545E+06	6.406
18	21	0.151309	1.815699	12.241293	2.551E+06	6.407
19	22	0.068239	2.259326	9.344100	2.545E+06	6.406
20	23	0.240410	1.673872	8.754734	2.551E+06	6.407
21	24	0.113568	3.034672	16.360465	2.551E+06	6.407
22	25	0.113764	2.802922	16.254037	1.020E+07	7.009
23	26	0.113807	2.637093	16.391239	1.020E+07	7.009
24	27	0.113587	1.864386	16.341227	1.020E+07	7.009

Fuente: Elaboración propia.

La Figura 1 muestra los valores de esparcidad obtenidos para cada una de las 24 capas convolucionales analizadas. Los valores oscilan entre 0.044 (capa 4) y 0.240 (capa 23), con un valor promedio de 0.109. Se observan fluctuaciones entre capas consecutivas dentro de cada etapa arquitectónica, sin una tendencia de crecimiento o decrecimiento sostenido a lo largo de toda la red.

Figura 1

Esparsidad por capa pesos entrenados de YOLO v1

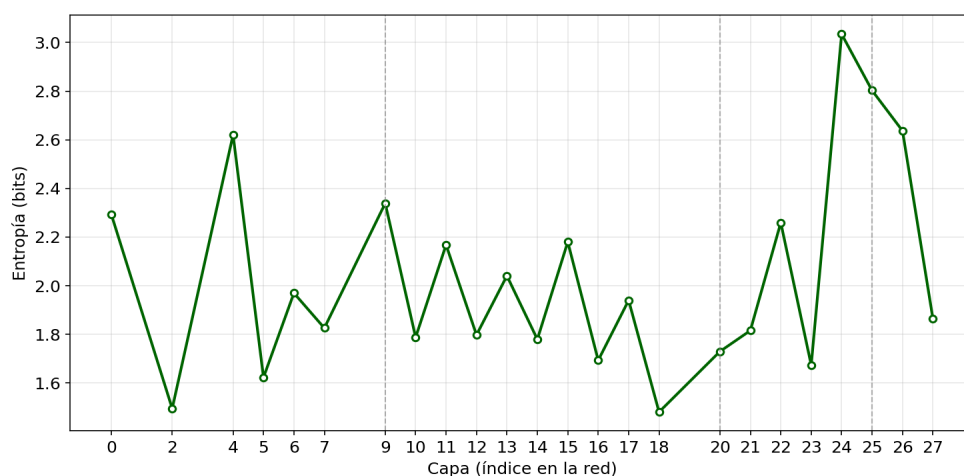


Fuente: Elaboración propia.

La Figura 2 presenta los valores de entropía calculados para cada capa. Los valores registrados varían entre 1.481 bits (capa 18) y 3.035 bits (capa 24), con un promedio de 2.038 bits. Al igual que en la Figura 1, no se observa un patrón monótono a lo largo de la profundidad de la red.

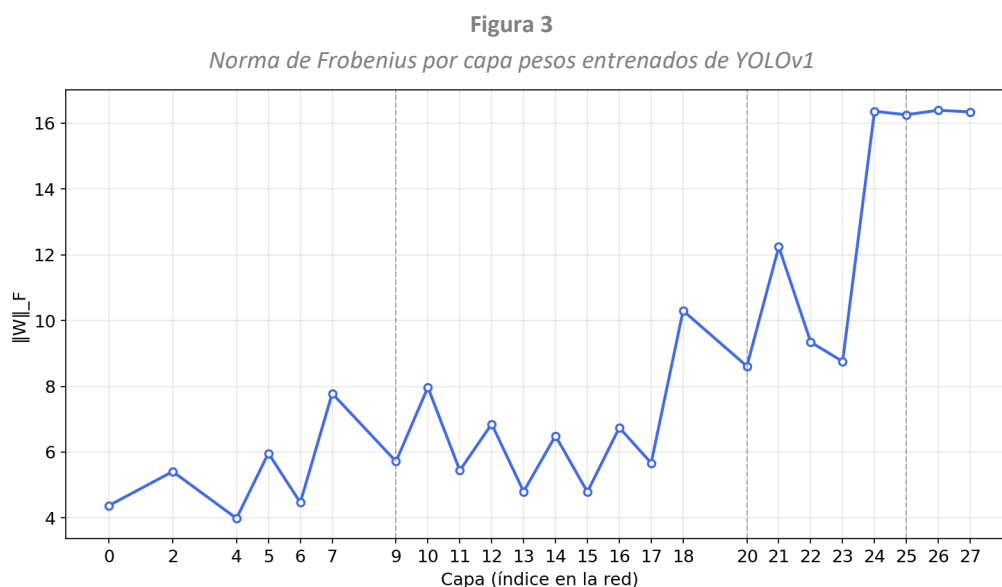
Figura 2

Entropía (H) de la distribución de pesos por capa



Fuente: Elaboración propia.

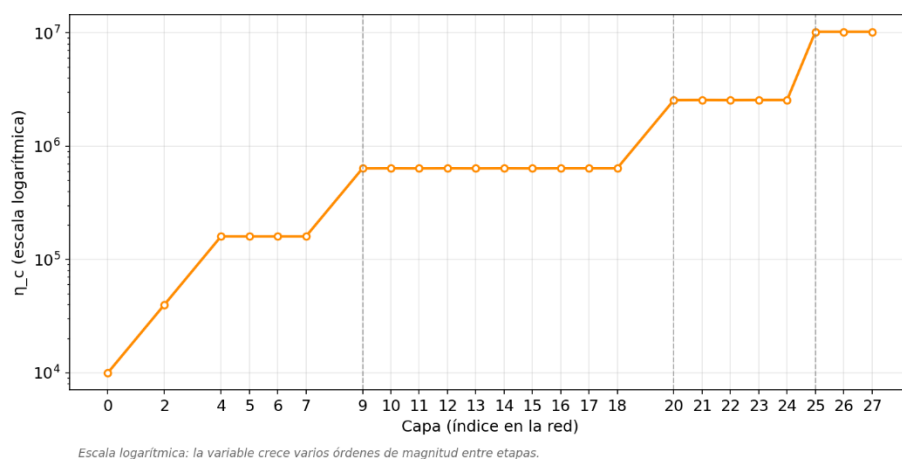
La Figura 3 muestra la Norma de Frobenius calculada para cada capa, con valores que van desde 3.986 (capa 4) hasta 16.391 (capa 26). A diferencia de las métricas anteriores, esta variable presenta un incremento progresivo a medida que aumenta el índice de capa, con saltos observables en las capas 9, 20 y 25.



Fuente: Elaboración propia.

La Figura 4 presenta los valores de η_c en escala logarítmica, debido al amplio rango de valores registrados (9.97×10^3 en la capa 0 hasta 1.02×10^7 en las capas 25 a 27). Los datos muestran un patrón escalonado, donde η_c se mantiene en valores similares dentro de un mismo rango de capas y presenta incrementos abruptos en las capas 9, 20 y 25.

Figura 4
Eficiencia Paramétrica por capa pesos entrenados de YOLO v1



Fuente: Elaboración propia.

La Tabla 3 presenta las estadísticas descriptivas de las métricas calculadas para las 24 capas convolucionales.

Tabla 3
Resumen estadístico de las métricas calculadas

ESTADÍSTICO	CAPA	SPARSITY	ENTROPÍA	NORMA DE FROBENIUS	EFICIENCIA (H_c)
Media	14.458	0.111	2.035	8.378	2.10×10^6
Desv. Estándar	8.016	0.048	0.415	4.150	3.25×10^6
Mínimo	0.000	0.044	1.481	3.986	9.97×10^3
Percentil 25	8.500	0.069	1.767	5.434	5.17×10^5
Mediana	14.500	0.114	1.902	6.799	6.38×10^5
Percentil 75	21.250	0.140	2.268	9.583	2.55×10^6
Máximo	27.000	0.240	3.035	16.391	1.02×10^7

Fuente: Elaboración propia.

Las Tablas 4 y 5 muestran los resultados completos del análisis de correlación de Pearson y Spearman, respectivamente.

Tabla 4
Análisis de Correlación de Pearson

VARIABLE A	VARIABLE B	COEFICIENTE (R)	P-VALOR	SIGNIFICATIVO (P < 0.05)	FUERZA
Capa	log(Eficiencia)	0.944	< 0.001	Sí	Muy fuerte
Capa	Norma de Frobenius	0.812	< 0.001	Sí	Fuerte
Norma de Frobenius	log(Eficiencia)	0.781	< 0.001	Sí	Fuerte
Sparsity	Entropía	-0.494	0.014	Sí	Moderada
Entropía	Norma de Frobenius	0.378	0.069	No	Débil
Capa	Sparsity	0.338	0.106	No	Débil
Sparsity	Norma de Frobenius	0.309	0.141	No	Débil
Entropía	log(Eficiencia)	0.271	0.200	No	Débil
Sparsity	log(Eficiencia)	0.247	0.245	No	Débil
Capa	Entropía	0.233	0.274	No	Débil

Fuente: Elaboración propia.

Tabla 5
Análisis de Correlación de Pearson

VARIABLE A	VARIABLE B	COEFICIENTE (P)	P-VALOR	SIGNIFICATIVO (P < 0.05)	FUERZA
Capa	Eficiencia	0.965	< 0.001	Sí	Muy fuerte
Norma de Frobenius	Eficiencia	0.852	< 0.001	Sí	Fuerte
Capa	Norma de Frobenius	0.828	< 0.001	Sí	Fuerte
Sparsity	Entropía	-0.691	< 0.001	Sí	Moderada
Sparsity	Norma de Frobenius	0.504	0.012	Sí	Moderada
Sparsity	Eficiencia	0.306	0.146	No	Débil
Capa	Sparsity	0.303	0.151	No	Débil
Entropía	Eficiencia	0.169	0.429	No	Muy débil
Capa	Entropía	0.157	0.463	No	Muy débil
Entropía	Norma de Frobenius	-0.005	0.981	No	Muy débil

Fuente: Elaboración propia.

La Tabla 6 resume únicamente los pares de variables que exhibieron correlaciones estadísticamente significativas, facilitando la identificación de las relaciones más relevantes entre las propiedades de las capas.

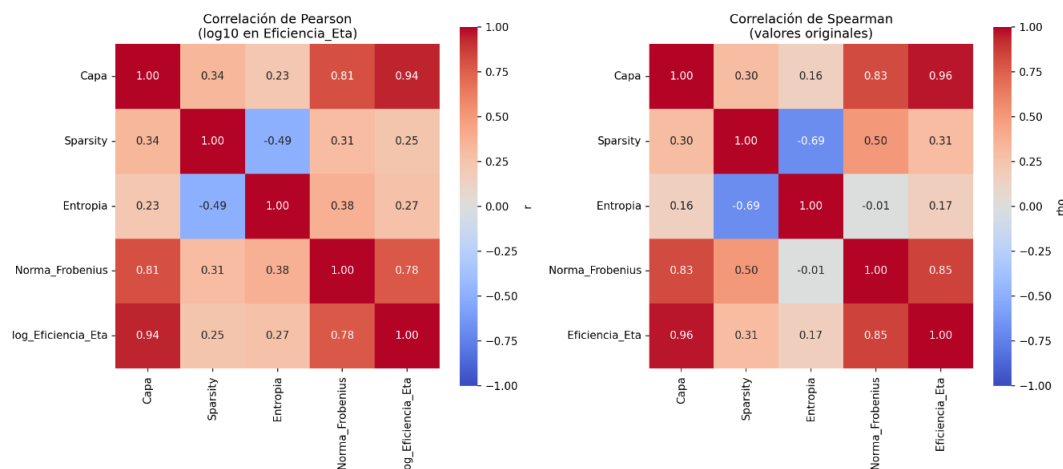
Tabla 6
Análisis de Correlación de Pearson

VARIABLE A	VARIABLE B	PEARSON (R)	SPEARMAN (P)	DIRECCIÓN	FUERZA (SPEARMAN)
Capa	Eficiencia	0.944***	0.965***	Positiva	Muy fuerte
Norma de Frobenius	Eficiencia	0.781***	0.852***	Positiva	Fuerte
Capa	Norma de Frobenius	0.812***	0.828***	Positiva	Fuerte
Sparsity	Entropía	-0.494*	-0.691***	Negativa	Moderada
Sparsity	Norma de Frobenius	0.309	0.504*	Positiva	Moderada

Fuente: Elaboración propia.

Para examinar las relaciones estadísticas entre las métricas calculadas (Sparsity, Entropía, Norma de Frobenius y Eficiencia), se generaron matrices de correlación utilizando los coeficientes de Pearson y Spearman. La Figura 5 presenta los mapas de calor correspondientes a ambos métodos:

Figura 5
Mapas de calor correlación de Pearson y Spearman



Fuente: Elaboración propia.

4. DISCUSIÓN

El análisis cuantitativo de los pesos de YOLOv1 revela una arquitectura con dinámicas estructurales no lineales y una jerarquía de importancia paramétrica claramente diferenciada. Al contrastar las métricas calculadas por capa con los coeficientes de correlación (Tablas 4, 5 y 6), se identifica que YOLOv1 no distribuye su carga computacional de forma uniforme, sino que la delega de manera ascendente hacia las capas finales.

Análisis de las Dinámicas Estructurales

Los resultados muestran una progresión crítica en la **Norma de Frobenius**, que asciende desde 3.986 (capa 4) hasta 16.391 (capa 26). Esta tendencia, validada por una correlación fuerte con la profundidad de la red ($r = 0.812$), indica que la red concentra su "energía" o relevancia semántica en las capas de decisión final. Paralelamente, el incremento logarítmico de la **Eficiencia** (η_c), que alcanza su pico (1.02×10^7) en el bloque terminal (capas 25-27), confirma un diseño arquitectónico donde el costo computacional está íntimamente ligado a la profundidad, reforzando la idea de una red que "aprende" a dedicar más recursos a las capas responsables de la predicción final.

Un hallazgo crítico reside en la relación entre **Sparsity y Entropía**. La correlación negativa significativa ($\rho = -0.691$) demuestra que, a medida que la redundancia (Sparsity) aumenta en una capa, su capacidad para codificar información diversa (Entropía) se ve comprometida. Por ejemplo, en la **Capa 23**, la alta esparsidad (0.240) se traduce en una entropía contenida (1.673), lo que evidencia un fenómeno de saturación: el modelo retiene conexiones de baja relevancia que no contribuyen a la riqueza informativa de la representación.

5. CONCLUSIONES

A partir de la triangulación de los datos y el análisis estadístico, se establecen las siguientes conclusiones:

1. **Jerarquía de Importancia:** Se confirma una dependencia directa entre la profundidad de la red y la magnitud de los pesos (Norma de Frobenius). YOLOv1 delega la complejidad representacional en las capas profundas, las cuales exhiben simultáneamente la mayor eficiencia y relevancia técnica.

2. **Redundancia Paramétrica:** Se ha identificado una "ineficiencia estructural" en las capas con alta esparsidad y baja entropía. La correlación negativa entre estas variables sugiere que el modelo no ha optimizado completamente sus recursos, manteniendo redundancias que no aportan valor semántico, siendo la **Capa 23** un claro ejemplo de este desbalance.
3. **Marco de Diagnóstico:** La combinación de la Norma de Frobenius y la Entropía de Shannon constituye un marco robusto para auditar redes neuronales. Este enfoque permite identificar "capas críticas" (de alta norma) frente a "capas prescindibles" (de alta esparsidad), proporcionando una hoja de ruta para futuras técnicas de **poda (*pruning*)** o compresión de modelos, capaces de reducir el tamaño del archivo .weights sin sacrificar la precisión en el reconocimiento.

En síntesis, este estudio demuestra que la arquitectura de YOLOv1, aunque efectiva, contiene una arquitectura subyacente con márgenes claros de optimización. El análisis demuestra que la arquitectura podría ser más ligera si se eliminan los parámetros que, estadísticamente, colapsan la entropía del modelo, validando que el estudio de la distribución de pesos es fundamental para la evolución de modelos de visión artificial más eficientes.

6. REFERENCIAS BIBLIOGRÁFICAS

1. **Redmon J, Divvala S, Girshick R, Farhadi A.** You Only Look Once: Unified, Real-Time Object Detection. En: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 779-788.
2. **Girshick R, Donahue J, Darrell T, Malik J.** Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. En: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2014. p. 580-587.
3. **Girshick R. Fast R-CNN.** En: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2015. p. 1440-1448.
4. **Shannon CE.** A Mathematical Theory of Communication. Bell Syst Tech J. 1948;27(3):379-423.
5. **Golub GH, Van Loan CF.** Matrix Computations. 4a ed. Baltimore: Johns Hopkins University Press; 2013.
6. **Blalock D, Ortiz JJ, Frankle J, Gutttag J.** What is the State of Pruning for Machine Learning? En: Proceedings of Machine Learning and Systems (MLSys); 2020. p. 129-146.