Análisis Comparativo de Graph Neural Networks: GCN vs MLP en Clasificación de Nodos

Jordi Blasco Lozano[1]

Resumen

Práctica 1 Graph Neural Networks (GNNs). Se ha investigado el comportamiento de las GNNs frente a modelos tradicionales (MLP) utilizando tanto un dataset sintético diseñado específicamente para aislar el valor de la estructura del grafo, como datasets de referencia (Cora y Citeseer). Los experimentos demuestran que las GCN superan dramáticamente a los MLP (99.9% vs 32.2% de accuracy) en escenarios donde las características de los nodos son ruidosas pero la homofilia es alta, confirmando que el mecanismo de message passing es crucial para recuperar la señal subyacente.

1. Introducción

En esta práctica, nos centramos en la tarea de **clasificación de nodos semi-supervisada**, donde solo una pequeña fracción de los nodos tiene etiquetas conocidas y debemos predecir el resto. El objetivo central es evidenciar las limitaciones de los modelos que ignoran la estructura relacional, como el Perceptrón Multicapa (MLP), y contrastarlos con modelos mejor diseñados para operar sobre grafos, específicamente las Graph Convolutional Networks (GCN).

La hipótesis fundamental que validaremos es que el mecanismo de *message passing* de las GCN permite mitigar el ruido en las características individuales mediante la agregación de información del vecindario, un proceso que es imposible para un MLP que asume independencia entre muestras.

2. Creación del Dataset Sintético

Para el análisis, no basta con usar datasets estándar donde no controlamos las propiedades de los datos. Se ha diseñado y generado el (*Custom Dataset*) con el objetivo de demostrar las diferencias fundamentales entre los modelos.

Generación de la Estructura: Se utilizó el modelo **Stochastic Block Model** para generar un grafo con las siguientes características:

- **Nodos:** 2000 nodos divididos en 4 comunidades del mismo tamaño.
- **Homofilia Estructural:** Se definió una probabilidad de conexión intra-clase $p_{in} = 0.02$ y una inter-clase $p_{out} = 0.001$. Esto asegura que, topológicamente, los nodos de la misma clase estén densamente conectados, creando comunidades bien definidas.

Generación de Características: Se generaron los vectores de características $\mathbf{x} \in \mathbb{R}^{32}$. Se utilizaron los mismos valores de señal y ruido del enunciado de la practica. Las características se generan utilizando la siguiente función.

$$\mathbf{x}_i = \alpha \cdot \mathbf{centroide}_{y_i} + \mathcal{N}(0, \sigma^2)$$

Donde la señal de la clase (α) es débil en comparación con la magnitud del ruido gaussiano (σ^2 alto).

Si las características fueran perfectas, un MLP obtendría casi 100% de acierto y la estructura del grafo sería irrelevante. Al introducir mucho ruido, forzamos al modelo a depender de la estructura: un modelo debe tener en cuenta a los vecinos para desambiguar la clase del nodo. Esto simula escenarios reales donde la información local es imperfecta.

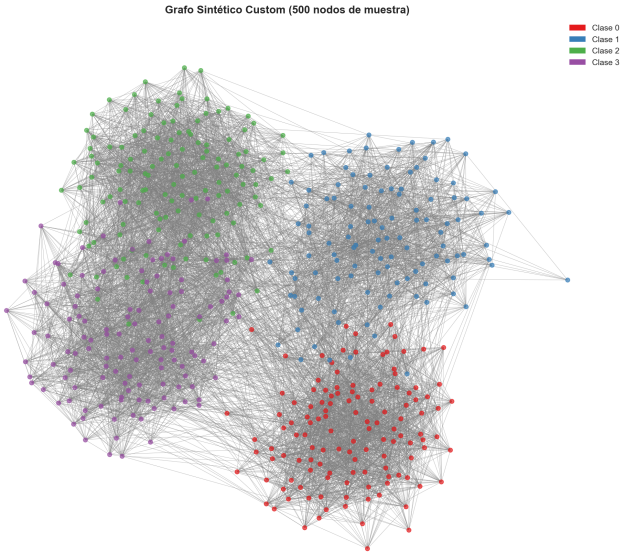


Figure 1. Visualización de la estructura del grafo sintético generado (SBM). Se observan claramente las 4 comunidades densamente conectadas internamente.

Tal como se aprecia en la Figure 1, la topología resultante es muy informativa. A pesar del ruido en los nodos individ-

uales, un nodo está casi siempre conectado a nodos de su misma clase, lo cual es la premisa que explota la GCN.

3. Modelos Implementados

3.1. Multi-Layer Perceptron (MLP)

Se implementó como una red feedforward con:

- Capa de entrada: Proyección lineal de dimensiones de entrada a ocultas.
- Función de activación: ReLU.
- Dropout: $p = 0.5$ para regularización.
- Capa de salida: Proyección lineal al número de clases.

Matemáticamente, para un nodo i, la salida depende exclusivamente de $\mathbf{x}_i$. Este modelo es ciego a la matriz de adyacencia $\mathbf{A}$.

3.2. Graph Convolutional Network (GCN)

Cada capa realiza la operación:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$$

Donde $\tilde{\mathbf{A}}$ es la matriz de adyacencia con self-loops. Esta operación realiza una suma ponderada de las características del propio nodo y las de sus vecinos, seguido de una transformación lineal y una no-linealidad. En nuestro contexto de ruido gaussiano, este promedio local reduce la varianza del ruido en un factor proporcional al grado del nodo (por la Ley de los Grandes Números), permitiendo que emerja la señal del centroide de la clase.

4. Resultados y Análisis

Se realizaron experimentos utilizando validación cruzada con 10 ejecuciones independientes para garantizar unos resultados robustos.

4.1. Rendimiento en Dataset Custom

Los resultados en el dataset sintético son contundentes y confirman nuestra hipótesis de diseño.

Table 1. Resultados en Dataset Custom (10 runs)

MODELO	ACC MEDIA	STD DEV	GAP
MLP	0.3223	$\pm$ 0.0175	-
GCN	**0.9998**	$\pm$ 0.0007	+67.7%

Análisis del colapso del MLP (0.322): Dado que hay 4 clases, un clasificador aleatorio obtendría un 0.25 (25%). El MLP con 0.32 apenas supera el azar. Esto indica que el ruido introducido en las características es tan alto que destruye casi toda la información discriminativa a nivel individual. El MLP no puede ver más allá del ruido de cada feature aislada.

Análisis del éxito de la GCN (0.999): La GCN alcanza la perfección. A pesar de que cada nodo individualmente es ruidoso e inclasificable, el colectivo (la comunidad) contiene la información perfecta. La topología SBM que creamos actúa como un mecanismo de corrección de errores extremadamente potente. La GCN ha aprendido a suavizar las características usando la estructura, recuperando la señal original casi intacta.

4.2. Validación en Benchmarks Reales

Para verificar que estas conclusiones no son un artefacto de nuestro dataset sintético, evaluamos los modelos en redes de citación reales: **Cora** y **Citeseer**.

Table 2. Resultados en Benchmarks Reales

	CORA		CITESEER	
MODELO	ACC	GAP	ACC	GAP
MLP	0.560	-	0.549	-
GCN	**0.807**	+24.7%	**0.690**	+14.1%

Disminución del Gap: Aunque la GCN sigue siendo muy superior (+24.7% en Cora), el MLP ya no falla estrepitosamente (0.56 vs 0.32). Esto se debe a que, en Cora, las características son palabras (Bag-of-Words). La presencia de palabras técnicas específicas (ej: "genoma") es por sí misma un predictor fuerte de la clase, independientemente de las citas. Sin embargo, la GCN en Cora logra un 80% al combinar esta información semántica con la información de citación (quién cita a quién). También hay que recalcar que la GCN, pese a tener un rendimiento decente en ambos datasets, no llega a la perfección de nuestro dataset sintético. Y esto se debe a que los grafos reales presentan una homofilia imperfecta y una estructura topológica más compleja. A diferencia del SBM, que impone comunidades matemáticamente segregadas, las redes de citación reales contienen enlaces transversales y fronteras difusas entre temas que limitan la capacidad de separación perfecta mediante la propagación de mensajes.

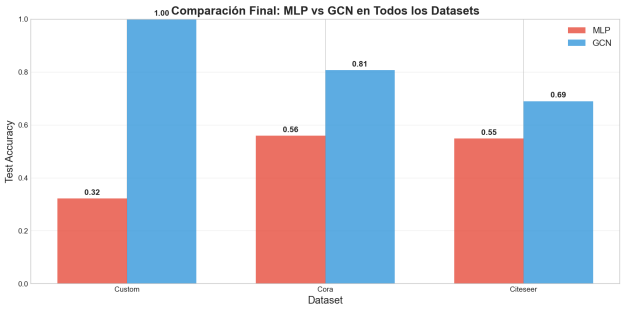


Figure 2. Comparativa final de Accuracy en los tres datasets evaluados.

La Figure 2 resume estos hallazgos, mostrando que aunque

la magnitud de la mejora varía según la calidad de las features (máxima en Custom, moderada en Cora/Citeseer), la GCN es sistemáticamente superior.

4.3. Impacto de Hiperparámetros y Entrenamiento

Curvas de Aprendizaje: Observamos que la GCN converge mucho más rápido y a pérdidas mucho menores. El MLP se estanca prematuramente en un valle de error alto, incapaz de optimizar más allá debido a la inconsistencia de las etiquetas respecto a las características ruidosas.

Figure 3. Curvas de entrenamiento (pérdida y accuracy). La GCN (naranja) alcanza rápidamente la convergencia, mientras el MLP (roja) no logra aprender.

En la Figure 3 se hace evidente la disparidad en la dinámica de aprendizaje. Mientras la loss de la GCN cae a cero, la del MLP permanece alta, indicando que el modelo no encuentra patrones fiables en las características aisladas. Podemos observar como vemos un comportamiento extraño en el plot de training, vemos como pese a saber que MLP es un modelo malo para nuestro dataset, la curva roja sube en esta gráfica a mas de 0.8 mientras que en la gráfica de validación se mantiene en todo momento al rededor del 0.3, esto se debe a que el modelo está experimentando un **sobreajuste (overfitting) masivo**. El MLP, al no poder apoyarse en la estructura para cancelar el ruido, opta por memorizar el ruido específico de las muestras de entrenamiento para reducir la pérdida. Sin embargo, dado que este ruido es aleatorio y no contiene información generalizable, la exactitud en validación permanece estancada, revelando que el modelo no ha aprendido ninguna característica discriminativa real.

Análisis de Hiperparámetros: Para validar la robustez de los modelos, he extendido la búsqueda de hiperparámetros a los tres datasets (Custom, Cora, Citeseer).

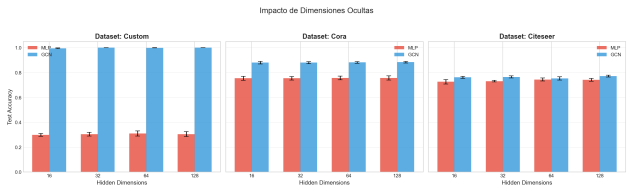


Figure 4. Impacto de las dimensiones ocultas en los tres datasets (Accuracy).

Dimensiones Ocultas (Figure 4): Al analizar las dimensiones [16, 32, 64, 128], la evidencia muestra un **estancamiento del rendimiento** más que una mejora progresiva. Pasar de 32 a 64 o 128 neuronas apenas varía el accuracy (se mantiene estable en $\approx 88\%$ en Cora). Esto indica que la complejidad del problema de clasificación satura rápidamente la capacidad del modelo; añadir más parámetros no permite extraer más información porque el límite viene dado por la calidad de los datos y la estructura, no por la potencia de cálculo. 16 dimensiones resultan suficientes para capturar la semántica latente.

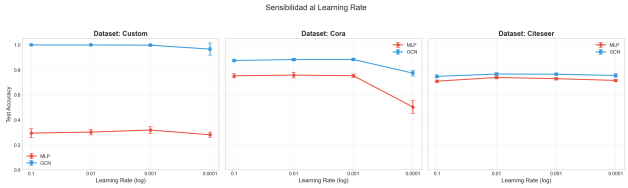


Figure 5. Sensibilidad al Learning Rate.

Learning Rate (Figure 5): El LR de 0.01 se confirma como el punto óptimo. La caída drástica observada en 0.0001 (especialmente en Cora con GCN bajando al 77.5%) se debe a una convergencia insuficiente. Con pasos de actualización tan pequeños y un número de épocas fijo (200), el optimizador no logra recorrer la superficie de error hasta el mínimo global, quedándose atascado en zonas subóptimas. En Cora, cuya superficie de optimización es más irregular que en el dataset sintético, este efecto se acentúa porque el modelo necesita actualizaciones más agresivas para escapar de los mínimos locales iniciales.

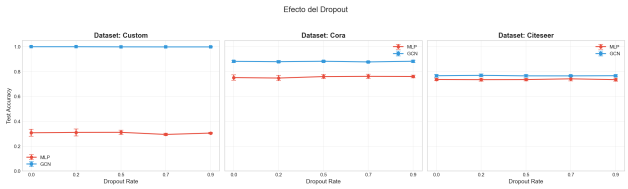


Figure 6. Efecto del Dropout.

Regularización - Dropout (Figure 6): El Dropout es una técnica que consiste en apagar aleatoriamente un porcentaje de neuronas durante el entrenamiento para evitar que coadapten y memoricen ruido. Sorprendentemente, la GCN mantiene un rendimiento extremadamente estable (88% en Cora) incluso con dropouts agresivos. La razón de este fenómeno es la **redundancia estructural**: aunque anulemos características internas de un nodo, el mecanismo de agregación (message passing) recupera información similar promediando a los vecinos. El grafo actúa como una copia de seguridad distribuida de la información, haciendo al modelo inusualmente robusto a la eliminación de características individuales.

Regularización - Weight Decay (Figure 7): El Weight De-

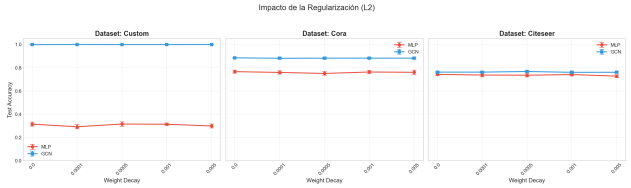


Figure 7. Impacto del Weight Decay (L2).

cay aplica una penalización L2 a los pesos ($\lambda||w||^2$) para mantenerlos pequeños y evitar funciones de decisión excesivamente complejas. Al igual que con el dropout, observamos que no importa demasiado el valor que pongamos, los resultados se mantienen planos. Esto sugiere que las GCN, por su naturaleza de promediado local (filtro pasobajo), son intrínsecamente regulares. La propia operación de convolución ya suaviza la señal y previene cambios bruscos, actuando como una regularización implícita potente, lo que hace que la regularización explícita (L2) tenga un impacto marginal comparado con modelos no estructurados.

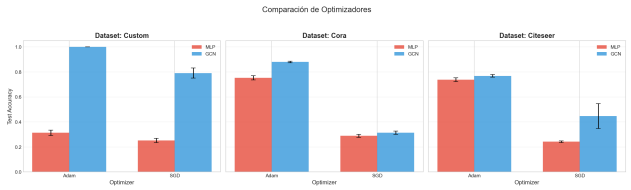


Figure 8. Comparativa de Optimizadores.

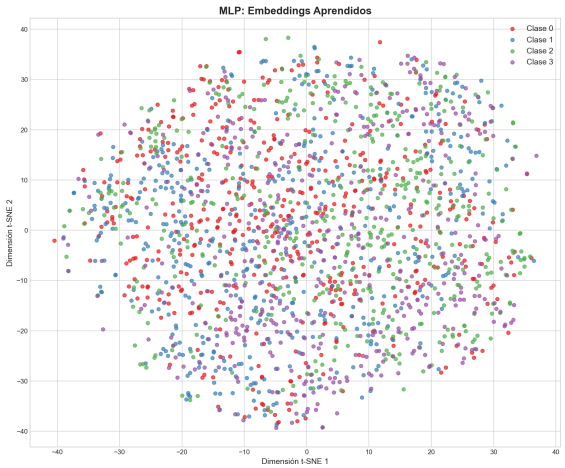
Optimizadores (Figure 8): Para entender los resultados, primero definamos brevemente los optimizadores a utilizar:

- **SGD:** Descenso de gradiente estocástico básico, actualiza pesos en dirección opuesta al gradiente.
- **Adam:** Combina inercia (promedio de gradientes pasados para suavizar dirección) con adaptación (normalización por magnitudes históricas), ajustando tasa y estabilidad de forma autónoma.

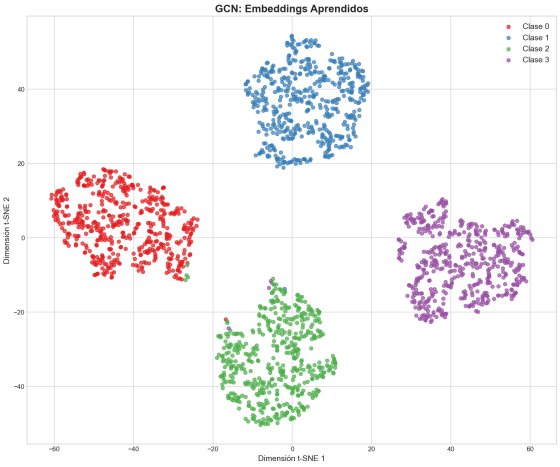
Los resultados muestran como SGD fracasa estrepitosamente en los datasets reales (Cora 30%, Citeseer 26%), comportándose casi como el azar. Sin mecanismos adaptativos ni momentum, SGD es incapaz de navegar la geometría compleja y dispersa de los grafos, quedándose atascado inmediatamente. Por el contrario, Adam logra converger eficazmente a valores altos (88% y 76%), demostrando que para entrenar GNNs de forma fiable, el uso de optimizadores adaptativos que gestionen la escala de los gradientes es una muy buena práctica.

5. Espacio Latente (t-SNE)

La visualización de los embeddings aprendidos mediante t-SNE proporciona la evidencia más intuitiva de la capacidad de la GCN para estructurar el espacio latente.



(a) Embeddings MLP



(b) Embeddings GCN

Figure 9. Visualización t-SNE del espacio latente aprendido por ambos modelos en el dataset sintético.

- **MLP (Figure 9(a)):** Muestra una nube dispersa donde las clases están severamente mezcladas (puntos de distintos colores superpuestos). Dado que el modelo solo ve características ruidosas, no logra proyectar los datos a un espacio donde sean linealmente separables.
- **GCN (Figure 9(b)):** Muestra clústeres extremadamente compactos y bien separados, con márgenes amplios entre clases. La estructura del grafo ha actuado como una fuerza de cohesión, obligando a los nodos conectados a tener representaciones vectoriales similares, limpiando efectivamente el ruido.

6. Conclusiones

Esta práctica ha servido para deconstruir el funcionamiento de las Graph Neural Networks.

1. Hemos demostrado que la información estructural es un recurso tan valioso como la información de características. Ignorarla, como hace el MLP, es desperdiciar la mitad de los datos disponibles.
2. El dataset sintético probó que la GCN es extraordinariamente robusta al ruido de entrada (0.99 de acierto vs 0.32), comportándose como un potente filtro de denoising estructural.
3. En datos reales, aunque las características sean informativas, la GCN proporciona una mejora crítica (+14-25%) al modelar el contexto relacional.

Como conclusión final, las GCN no son solo otra arquitectura, sino un cambio de paradigma: pasamos de aprender de entidades aisladas a aprender de sistemas interconectados.

[1]Jordi Blasco Lozano – DNI: 74527208D – jordiblloz@gmail.com