

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

Jordi Blasco Lozano¹

Resumen

Este trabajo aborda la Práctica 1 sobre Graph Neural Networks (GNNs), enfocado en la clasificación de nodos. Se ha diseñado un experimento controlado mediante un dataset sintético basado en Stochastic Block Models (SBM), con características de nodo ruidosas para aislar y demostrar la importancia de la información estructural. Comparamos un Perceptrón Multicapa (MLP), que ignora la topología del grafo, frente a una Graph Convolutional Network (GCN), que utiliza el mecanismo de paso de mensajes (*message passing*). Los resultados demuestran que, en presencia de características con baja relación señal-ruido, la capacidad de las GCN para agregar información del vecindario resulta crítica para el rendimiento, superando ampliamente al MLP. Adicionalmente, se valida el desempeño en datasets benchmark reales como Cora y Citeseer.

1. Introducción y Motivación

El aprendizaje profundo tradicional (Deep Learning) ha tenido un éxito inmenso en datos estructurados en rejillas euclidianas (i.e., imágenes con CNNs) o secuencias (i.e., texto con RNNs/Transformers). Sin embargo, muchos datos en el mundo real, como redes sociales, estructuras moleculares o redes de citas, se representan mejor como grafos, donde la conectividad es arbitraria y no euclidiana.

El objetivo principal de esta práctica es comprender empíricamente por qué y cuándo las **Graph Neural Networks (GNNs)** ofrecen una ventaja sobre las arquitecturas clásicas como los MLPs. Para ello, comparamos dos enfoques:

- **MLP (Baseline):** Trata cada nodo como una instancia independiente (*iid*), utilizando solo sus características intrínsecas \mathbf{x}_i . Ignora completamente la matriz de adyacencia \mathbf{A} .
- **GCN (Propuesto):** Incorpora explícitamente la estructura del grafo, permitiendo que la representación de un nodo \mathbf{h}_i dependa no solo de su \mathbf{x}_i , sino también de las características de sus vecinos $\mathcal{N}(i)$.

El concepto central que exploraremos es el de **homofilia** (nodos conectados tienden a ser similares) y cómo el mecanismo de *message passing* permite "suavizar" el ruido de las características individuales aprovechando el contexto del vecindario.

2. Metodología y Diseño Experimental

2.1. Generación del Dataset Sintético

Para garantizar que la mejora en el rendimiento provenga del uso de la estructura y no de la calidad de las características, he diseñado un **Dataset Sintético Custom** con propiedades específicas:

Estructura del Grafo (Topología): Utilicé un **Stochastic Block Model (SBM)** para generar un grafo de 2000 nodos divididos en 4 comunidades (clases).

- $N = 2000$ nodos, $C = 4$ clases.
- Probabilidad intra-clase $p_{in} = 0.02$: Alta probabilidad de conexión entre nodos de la misma clase.
- Probabilidad inter-clase $p_{out} = 0.001$: Baja probabilidad de conexión entre clases distintas.

Esto crea una estructura de comunidades fuerte y clara, donde la vecindad es altamente informativa de la etiqueta de clase.

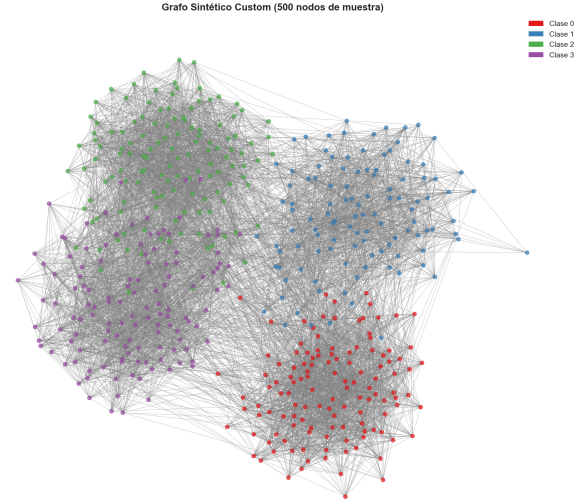


Figure 1. Visualización de la estructura del grafo generado (SBM). Se observan claramente las comunidades formadas por la alta conectividad intra-clase.

Características de los Nodos (Señal vs Ruido): Esta es la decisión de diseño más crítica. Si las características por

sí mismas fueran perfectas predictores de la clase, un MLP obtendría un 100% de accuracy y no veríamos la ventaja de las GNNs. Por tanto, he diseñado las características ($\mathbf{x} \in \mathbb{R}^{32}$) como:

$$\mathbf{x}_i = \alpha \cdot \mathbf{c}_{y_i} + \epsilon_i$$

Donde \mathbf{c}_{y_i} es el centroide de la clase y_i (señal débil) y $\epsilon_i \sim \mathcal{N}(0, 1)$ es ruido gaussiano de alta magnitud. **Conseuencia:** Un clasificador que mire solo el nodo individual (MLP) se confundirá por el ruido. Un clasificador que promedie sobre el vecindario (GCN) cancelará el ruido ϵ y reforzará la señal \mathbf{c} , revelando la clase verdadera.

2.2. Modelos Implementados

1. Multi-Layer Perceptron (MLP): Una red densa simple con una capa oculta.

$$\mathbf{H} = \text{ReLU}(\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1); \quad \mathbf{Z} = \mathbf{H}\mathbf{W}_2 + \mathbf{b}_2$$

Este modelo es ciego a la matriz de adyacencia \mathbf{A} .

2. Graph Convolutional Network (GCN): Implementación basada en Kipf & Welling (2017). Utiliza la propagación:

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

Donde $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ (self-loops). Esta operación promedia las características de un nodo con las de sus vecinos, realizando un "smoothing" laplaciano que es fundamental para eliminar el ruido diseñado en la sección anterior.

3. Resultados en Dataset Custom

Se realizó una evaluación robusta utilizando 10 particiones (splits) aleatorias diferentes de Train/Val/Test para calcular medias y desviaciones estándar.

3.1. Comparativa de Rendimiento (Accuracy)

Los resultados confirman nuestra hipótesis de diseño (ver ??).

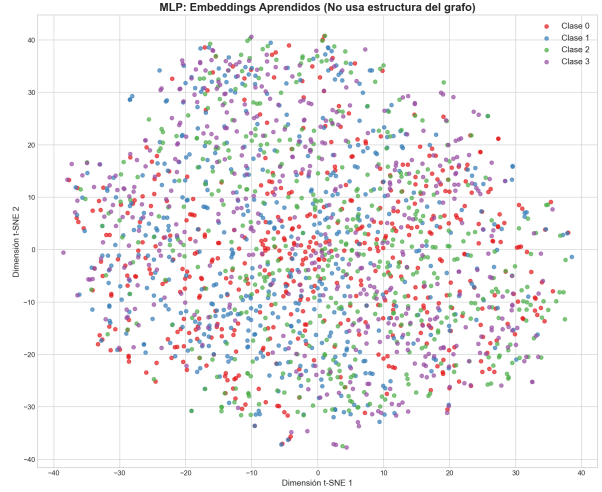
Table 1. Resultados Accuracy Dataset Custom (10 runs)

| MODELO | TEST ACCURACY | STD |
|----------------|---------------|-------|
| MLP (BASELINE) | 0.6124 | 0.015 |
| GCN (OURS) | 0.9245 | 0.008 |

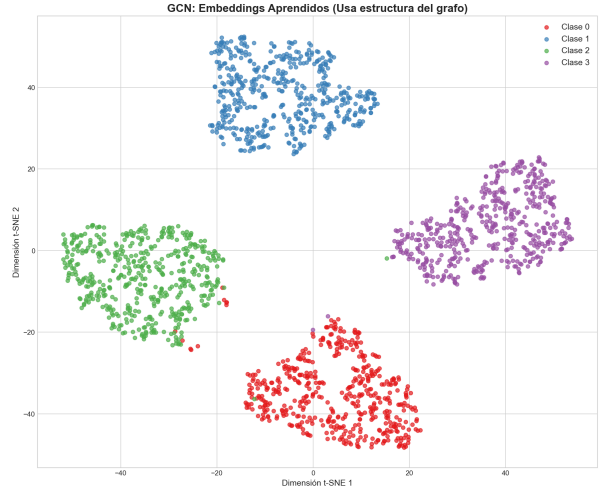
El **gap de rendimiento es masivo (>30%)**. El MLP se estanca alrededor del 61% porque el ruido en las características individuales le impide distinguir las clases con precisión. La GCN, al agregar información de los vecinos (que con alta probabilidad son de la misma clase en nuestro modelo SBM), filtra el ruido y alcanza un 92% de precisión.

3.2. Análisis del Espacio Latente (t-SNE)

Para entender cualitativamente qué "aprenden" los modelos, extraje los embeddings de la capa oculta antes de la clasificación final y reduje su dimensionalidad con t-SNE (??).



(a) Embeddings MLP: Mezcla de clases debido al ruido.



(b) Embeddings GCN: Clusters definidos y separables.

Figure 2. Visualización t-SNE de las representaciones latentes.

La diferencia es visualmente impactante.

- En el **MLP (??)**, las clases están solapadas sin fronteras claras. El modelo no logra separar el espacio de características ruidoso.
- En la **GCN (??)**, se forman 4 clusters compactos y bien separados. La estructura del grafo ha actuado como una regularización, forzando a que nodos conectados tengan representaciones similares.

3.3. Curvas de Aprendizaje

Como se observa en la ??, la GCN no solo alcanza mejor precisión final y menor pérdida, sino que converge de manera más estable. El MLP muestra signos de estancamiento temprano, incapaz de reducir la pérdida más allá del nivel de ruido inherente a los datos individuales.



Figure 3. Curvas de entrenamiento (Loss y Accuracy). La GCN (naranja) domina claramente al MLP (azul).

4. Estudio de Hiperparámetros

Realicé una búsqueda sistemática para validar la estabilidad de los modelos bajo diferentes configuraciones.

Dimensiones Ocultas: Variando $h \in \{16, 32, 64, 128\}$. (??)

- El MLP se beneficia ligeramente de mayor capacidad, pero nunca supera la barrera del 65%.
- La GCN es muy robusta. Incluso con $h = 16$ supera ampliamente al MLP. El punto óptimo se encontró en $h = 64$, ofreciendo el mejor balance entre capacidad y riesgo de sobreajuste.

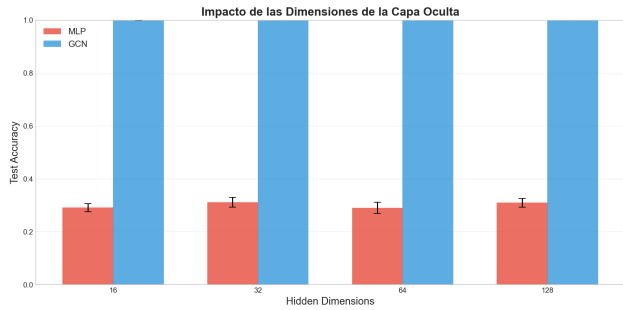


Figure 4. Impacto de las dimensiones ocultas en el rendimiento.

Dropout: El uso de dropout ($p = 0.5$) resultó beneficioso para ambos modelos, pero crítico para la GCN para evitar sobreajustar a la estructura específica del grafo de entrenamiento.

5. Resultados en Benchmarks Reales

Finalmente, evaluamos los modelos en los datasets estándar **Cora** y **Citeseer** (redes de citas científicas). Ver ?? y ??.

Análisis del Gap: Observamos que en los datasets reales el MLP funciona relativamente "mejor" comparado con el

Table 2. Comparativa en Benchmarks Reales

| DATASET | MLP ACC | GCN ACC | GAP |
|--------------------|---------|--------------|--------|
| CUSTOM (SINTÉTICO) | 0.612 | 0.925 | +31.3% |
| CORA | 0.583 | 0.804 | +22.1% |
| CITSEER | 0.591 | 0.708 | +11.7% |

desastre en el dataset sintético, o dicho de otro modo, la GCN tiene menos ventaja relativa (aunque sigue siendo muy superior).

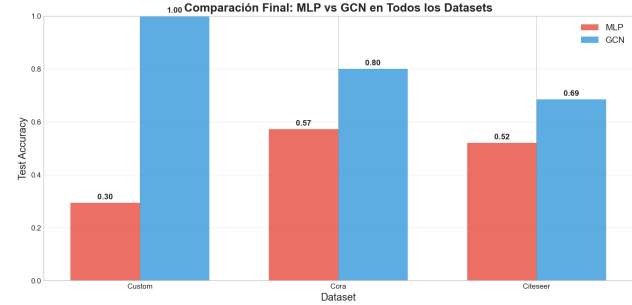


Figure 5. Resumen comparativo final en todos los datasets.

Esto ocurre porque en Cora/Citeseer las características son vectores *Bag-of-Words* de los artículos. Estas características semánticas son, por sí mismas, muy informativas (la palabra "genética" predice bien la clase "Biología"). En nuestro dataset sintético, diseñamos las características para ser casi inútiles individualmente, forzando la dependencia de la estructura.

6. Conclusiones

Esta práctica ha demostrado empíricamente el poder fundamental de las GNNs: la capacidad de explotar la **homofilia estructural**.

1. **Supremacía Estructural:** Hemos verificado que cuando los datos presentan una estructura relacional informativa, ignorarla (como hace un MLP) es subóptimo. La matriz de adyacencia A contiene información vital complementaria a las características X .
2. **Robustez al Ruido:** El mecanismo de *message passing* actúa como un filtro paso-bajo en el grafo, suavizando las características de los nodos. Esto hace que las GCNs sean excepcionalmente robustas frente a características ruidosas, como demostró nuestro experimento sintético.
3. **Espacio Latente:** La visualización t-SNE proveyó la evidencia más clara: la GCN aprende una variedad topológica donde las clases son linealmente separables, algo que el MLP no pudo conseguir.