Ismael Ruiz García

XARXES NEURONALS DE GRAFS PER A LA VALIDACIÓ DE PROPIETATS DE COMPONENTS QUÍMICS

TREBALL DE FI DE GRAU

dirigit per Dr. Francesc Serratosa Casanelles

Grau d'Enginyeria Informàtica

UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2024

Resum.

En aquest projecte es desenvolupa múltiples models predictius per estudiar molècules utilitzant la biblioteca MolGraph, convertint estructures moleculars en grafs per anàlisi en química computacional. S'inicia amb la base de dades QM7 i després s'adapta a la base de dades de la URV. Els resultats preliminars mostren que els models adaptats poden predir propietats moleculars amb alta precisió, evidenciant que la metodologia pot transferir-se i adaptar-se a diferents conjunts de dades, el que facilita avanços en la investigació i aplicacions en disseny assistit per computadora.

Resumen.

En este proyecto se desarrolla múltiples modelos predictivos para estudiar moléculas utilizando la biblioteca MolGraph, convirtiendo estructuras moleculares en grafos para análisis en química computacional. Se inicia con la base de datos QM7 y después se adapta a la base de datos de la URV. Los resultados preliminares muestran que los modelos adaptados pueden predecir propiedades moleculares con alta precisión, evidenciando que la metodología puede transferirse y adaptarse a diferentes conjuntos de datos, lo que facilita avances en la investigación y aplicaciones en diseño asistido por computador.

Abstract.

This project develops multiple predictive models to study molecules using the MolGraph library, converting molecular structures into graphs for computational chemistry analysis. It starts with the QM7 database and then adapts to the URV database. Preliminary results show that the adapted models can predict molecular properties with high accuracy, evidencing that the methodology can be transferred and adapted to different dataset, facilitating research advances and applications in computer-aided design.

Índex

Índex de taules

Índex de figures

Índex d'equacions

1 Introducció

La capacitat de predir propietats moleculars de manera precisa mitjançant models computacionals és un avanç crític que impulsa la innovació en diverses disciplines científiques, inclòs la química medicinal, el disseny de nou material i la bioinformàtica. En un moment en què el volum de dades biomoleculars creix exponencialment, la necessitat d'automatitzar i optimitzar tant la creació com la validació de models predictius és més imperiosa que mai. Els models predictius desenvolupats mitjançant tecnologies avançades d'aprenentatge automàtic, especialment els basats en arquitectures de grafs, han demostrat ser extremadament eficaços per manejar la complexitat i la gran dimensionalitat de les dades moleculars. Aquest document descriu un programa integral dissenyat tant per a la creació com per a l'avaluació de models d'aprenentatge profund, garantint la seva robustesa i fiabilitat abans de la seva implementació en entorns reals.

1.1 Context de l'estudi

El camp de la química computacional ha evolucionat de manera significativa amb la integració de l'aprenentatge automàtic, especialment mitjançant l'adopció de grafs que ofereixen un nou nivell de potència analítica. Aquests models no solament acceleren substancialment el procés de recerca i desenvolupament en àmbits com la farmacologia i l'enginyeria de material, sinó que també permet descobrir i modelar relacions complexes entre l'estructura molecular i les seves propietats funcional. L'aspecte distintiu d'aquest projecte rau en la seva doble finalitat: la creació i la validació de models. El desenvolupament de models precisos i la seva posterior validació són essencials, ja que proporcionen no només eines per a la predicció sinó també per a la confirmació de la seva eficàcia i aplicabilitat en contextos reals. La validació rigorosa, que inclou la capacitat del model per generalitzar a noves dades, és crucial per assegurar que els models predictius són aptes per a aplicacions d'avantguarda i descobriments innovadors en el camp. Aquest projecte aborda aquestes necessitats mitjançant un enfocament sistemàtic que combina la innovació en la creació de models amb estratègies robustes per a la seva avaluació.

1.2 Importància i justificació del projecte

El projecte té una importància crítica en l'era moderna de la química computacional, on la capacitat de predir ràpidament i amb precisió les propietats de nous compostos pot accelerar significativament els cicles d'innovació i desenvolupament en moltes disciplines científiques. El desenvolupament i la validació de models computacionals que poden simular i predir components moleculars amb alta fiabilitat són de vital importància. Aquests models no només poden reduir el temps i el cost associar als experiments de laboratori tradicionals, sinó que també augmenten la capacitat de descobrir noves substàncies amb propietats desitjables, com ara medicaments més efectius o materials amb propietats especials per a aplicacions industrials i tecnològiques.

La justificació del projecte rau en la necessitat emergent d'abordar i superar les limitacions actuals en la predicció de propietats químiques. Utilitzant arquitectures basades en grafs, aquest projecte pretén oferir una millora substancial sobre mètodes existent, proporcionant una aproximació més granular i contextual a l'anàlisi de dades moleculars. Així mateix, la validació d'aquests models en conjunts de dades diverses assegura que les prediccions siguin robustes i aplicables a una àmplia gamma de situacions reals, augmentant la seva utilització pràctica.

1.3 Objectiu del projecte

L'objectiu principal d'aquest projecte és desenvolupar i validar models predictius utilitzant dues bases de dades específiques: la base de dades QM7 i una base de dades proporcionada per la URV. Per això, s'emprarà la biblioteca MolGraph, que facilita la implementació de models computacionals avançats en el camp de la química computacional. Addicionalment, el projecte busca desenvolupar una eina basada en MolGraph que permet als investigadors crear, entrenar i validar els seus propis models predictius amb bases de dades personalitzades.

Els objectius específics del projecte són:

1. **Desenvolupament de models per a dues bases de dades diferents:**
    - Base de dades QM7: Utilitzar aquesta base de dades àmpliament reconeguda per a desenvolupar models predictius que demostren la capacitat de MolGraph per a manejar i analitzar dades moleculars complexes, establint un punt de referència per a l'eficàcia del modelatge.
    - Base de dades de la URV: Desenvolupar models predictius utilitzant la base de dades, que se centra en la predicció de l'afinitat molecular. Això permetrà avaluar com els models construïts amb MolGraph poden ser adaptats i afinats per predir propietats moleculars específiques, com l'afinitat, en nous conjunts de dades.
2. **Creació i distribució de scripts de desenvolupament i validació de models:**
    - Desenvolupar scripts de Python que automatitzin la creació, entrenament i validació de models utilitzant MolGraph i especialment centrats en la base de dades de la URV. Aquests scripts seran dissenyats per a ser intuïtius i accessibles, amb documentació completa.
3. **Validació dels models:**
    - Després del desenvolupament i la implementació dels models en les bases de dades QM7 i URV, es farà una anàlisi detallada dels resultats. Aquesta anàlisi inclourà la comparació del rendiment dels models entre ambdues bases de dades, destacant diferències, fortaleses i àrees de millora. Això proporcionarà una comprensió profunda de com cada model gestiona diferents tipus de dades moleculars i la seva eficàcia.

Aquest conjunt d'objectius busca avançar en les tècniques de modelatge predictiu utilitzant MolGraph en química computacional i fer accessible aquesta tecnologia a la comunitat científica, permetent a investigadors i desenvolupadors de tot el món beneficiar-se i contribuir a les últimes innovacions en el camp de manera pràctica i efectiva.

2 Aprenentatge profund

L'aprenentatge profund és una tècnica d'aprenentatge automàtic que emplena xarxes neuronals artificials amb múltiples capes de processament per modelar abstraccions complexes en dades. Aquesta tècnica, caracteritzada per la seva capacitat per aprendre automàticament les característiques òptimes per a la tasca en qüestió, ha revolucionat àrees com la investigació.

2.1 Definició d'aprenentatge profund

L'aprenentatge profund es defineix com l'ús de xarxes neuronals amb múltiples capes de processament, on cada capa transforma la seva entrada en un nivell més abstracte i representatiu que la seva predecessora. A través d'aquesta estructura de capes, les xarxes neuronals profundes poden capturar complexitats en les dades que van més enllà dels mètodes d'aprenentatge automàtic més tradicionals poden assolir.

2.1.1 Components d'una Neurona Artificial

Una neurona artificial és el bloc constructiu bàsic d'una xarxa neuronal i pot ser descrita matemàticament de la següent manera(Tello, 2018)):

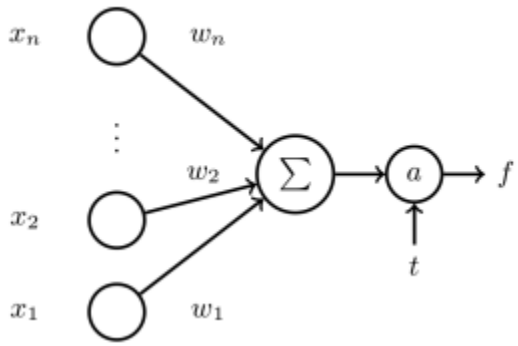


Figura 1. Representació d'una Neurona Artificial

- **Suma Ponderada(Z):** Cada neurona rep entrades x_1, x_2, ..., x_n que es multipliquen per pesos corresponents w_1, w_2, ..., w_n i es sumen juntament amb un terme de biaix b.

$$Z = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b \quad (1)$$

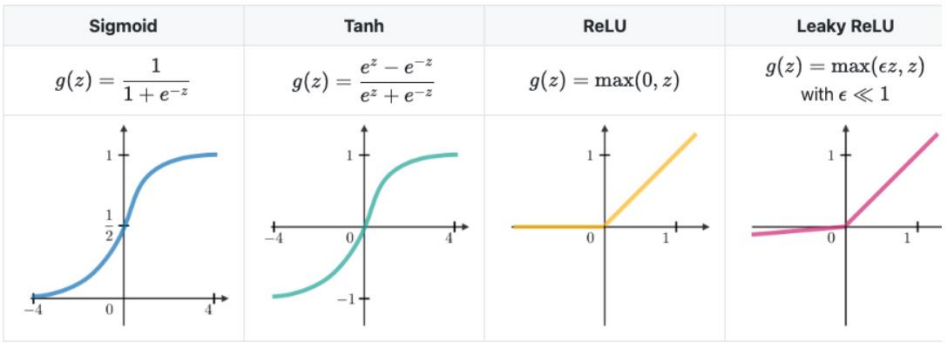


Figura 2. Funcions d'activació

- **Funció d'activació(σ):** La suma ponderada es passa a través d'una funció d'activació no lineal per a obtenir una sortida *a* de la neurona.

$$a = \sigma(Z) \quad \textbf{(2)}$$

Les funcions d'activació més comunes són la sigmoide, Tanh i ReLU. Aquestes funcions introdueixen no-linealitats que són crucials per aprendre i representar dades complexes.

2.1.2 Procés de Feedforward

El procés de Feedforward és el flux principal de càlcul en una xarxa neuronal, on la informació es mou exclusivament cap endavant, des de les capes d'entrada, a través de les capes ocultes, fins a la capa de sortida. Aquest procés és fonamental per a entendre com les xarxes neuronals processen informació i realitzen les prediccions (Mcinerney, 2023).

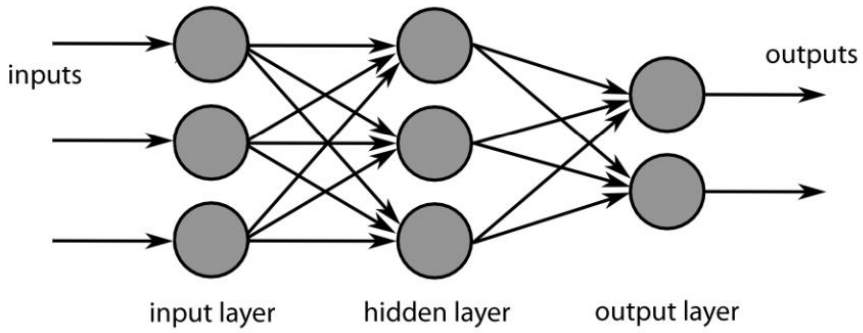


Figura 3. Representació d'una xarxa neuronal

2.1.2.1 Etapes del procés de Feedforward

1. **Capes d'entrada:**

Les neurones en la capa d'entrada actuen principalment com a receptors de les dades externes. Aquestes no realitzen cap mena de processament actiu sobre les dades, sinó que simplement les transmet a les capes subsegüents.

En alguns casos, la capa d'entrada pot incloure transformacions simples com la normalització o estandarditzación de les dades per facilitar l'aprenentatge més efectiu de capes següents.

2. **Capes Ocultes:**

En cada capa oculta, cada neurona rep una combinació lineal de les sortides de totes les neurones de la capa anterior. Matemàticament, per una neurona individual s'expressa com una suma ponderada d'aquestes entrades:

$$z_i^{(l)} = \sum_j W_{ij}^{(l)} a_j^{(l-1)} + b_i^{(l)} \quad \textbf{(3)}$$

On $z_i^{(l)}$ és l'input lineal a la neurona *i* en la capa *l*. $W_{ij}^{(l)}$ són els pesos que connecten la neurona *j* de la capa *l-1* amb la neurona *i* en la capa *l*. $a_j^{(l-1)}$ són les activacions de la capa anterior, i $b_i^{(l)}$ es el biaix per a la neurona *i*.

Després de calcular la suma ponderada, cada neurona aplica una funció d'activació no lineal a aquest valor. La funció d'activació introdueix la no-linealitat necessària que permet a la xarxa neuronal aprendre i modelar relacions complexes entre les dades d'entrada i sortida.

3. **Capa de sortida:**

En l'última capa, la capa de sortida, el procés es conceptualment el mateix que en les capes ocultes.

Durant el feedforward, cada capa calcula les seves activacions en seqüencia, començant amb la capa d'entrada i desplaçant-se successivament a través de cada capa oculta fins la capa de sortida. Aquest procés es realitza per a cada instància de les dades d'entrada al lot d'entrenament o al conjunt de proves.

3 Xarxes Neuronals de Grafs (GNNs)

3.1 Introducció a les Xarxes Neuronals de Grafs

Les xarxes neuronals de grafs (GNNs) són una classe avançada de models d'aprenentatge automàtica dissenyat per gestionar dades estructurades en forma de grafs. Aquesta estructura de dades es compon de nodes, que representen entitats, i arestes, que denoten les relacions entre aquestes entitats. Les GNNs son essencials per analitzar dades que no s'ajusten naturalment a formats tabulars o seqüencial, permetent una representació més rica i detallada de les relacions entre dades.

3.1.1 Que són els Grafs?

Un graf és una estructura matemàtica utilitzada per modelar pars de relacions entre objectes. Matemàticament, un graf *G* es defineix com un par ordenat *G = (V, E)* compost per (Grinberg, 2023):

- **Un conjunt de vèrtex o nodes V:** Aquests vèrtexs poden representar qualsevol entitat, com persones d'una xarxa social, estacions en un sistema de transport, o àtoms en una molècula.
- **Un conjunt d'arestes o arcs R:** Aquestes arestes són pars no ordenats (en grafs no dirigits) o pars ordenats (en grafs dirigits) de vèrtexs que representen la relació o connexió entre els nodes. Per exemple, un graf no dirigit, una aresta *(u, v)* indica una connexió bidireccional entre els vèrtexs *u* i *v*. Mentre que en un graf dirigit, l'aresta *(u, v)* representaria una relació de *u* i *v*, però no necessàriament en sentit invers.

Els grafs poden representar diversos tipus d'informació i relacions en nombrosos camps, des de les ciències socials fins a la bioinformàtica i la telemàtica, proporcionant un marc natural per modelar tant les relacions binàries com les xarxes més complexes.

3.1.2 Perquè Xarxes Neuronals per a Grafs?

L'aplicació de xarxes neuronals a estructures de grafs permet explotar profundament les relacions estructurals entre dades. Les GNNs són capaces d'aprendre com s'influeixen mútuament els nodes a través de les connexions, cosa que permet fer inferències i prediccions sofisticades sobre dades noves i no vistes basades en l'aprenentatge de patrons de connexió existents. Les GNNs implementen mecanismes d'agregació i d'actualització per combinar i actualitzar informació dels nodes i els seus veïns, permetent que l'aprenentatge en propagui a través de l'estructura del graf.

Aquesta capacitat de modelar interdependències complexes fa que les GNNs siguin úniques en comparació amb altres arquitectures d'aprenentatge automàtic, que poden no gestionar de manera nativa la data relacional o estructurada d'aquesta manera. La naturalesa flexible i potent de les GNNs les permet adaptar-se a una àmplia varietat de tipus de dades i mides de grafs, el que fa que siguin particularment útils en camps com la química computacional per a predir interaccions moleculars i propietats.

3.2 Tipus de Xarxes Neuronals de Grafs

3.2.1 GT

L'arquitectura del Graph Transformer consta de dues variants principals: la capa de Graph Transformer i la capa de Graph Transformer amb característiques d'arcs (Vijay Prakash Dwivedi, 2021). La primera està dissenyada per a grafs que no tenen atributs explícits en els seus arcs. Mentre que la segona manté un canal de característiques dels arcs designada per a incorporar la informació disponible en els arcs i mantenir les seves representacions abstractes en cada capa.

Capa de GT

Entrada

Primer, es separa les incrustacions de nodes i arcs d'entrada per passar-les a la capa de GT. Per un graf G amb característiques de node α_i per a cada node i i característiques de l'arc β_{ij} per a cada arc entre els node i i j, es passen les característiques del node α_i i les característiques de l'arc β_{ij} a través d'una projecció lineal per a incrustar-les en característiques ocultes de d dimensions h_i^0 *i* e_{ij}^0, respectivament.

Capa de Graph Transformer

La capa de Graph Transformer segueix de prop l'arquitectura de Transformer original proposada per (Vaswani, 2023). Es defineix les equacions de actualització de nodes per a una capa l, on es calcula una nova representació h_i^{l+1} per a cada node i com una suma ponderada de les representacions veïnals h_j^{l+1} amb els pesos calculats mitjançant 'Attention":

$$\hat{h}_i^{\ell+1} = O_h^{\ell} \Big\|_{k=1}^{H} \Big(\sum_{j \in \mathcal{N}_i} w_{ij}^{k,\ell} V^{k,\ell} h_j^{\ell} \Big),$$
$$\text{where, } w_{ij}^{k,\ell} = \text{softmax}_j \Big(\frac{Q^{k,\ell} h_i^{\ell} \cdot K^{k,\ell} h_j^{\ell}}{\sqrt{d_k}} \Big) \qquad \textbf{(4)}$$

On O_h^l és una matriu de sortida, H és el número de "heads Attention", $N(i)$ és el conjunt de nodes veïns de i, $Q^{k,l}$, $K^{k,l}$, $V^{k,l}$, són matrius de paràmetres, i d_k és la dimensió de les "heads Attentions". Després del "head Attention", les representacions actualitzades es passen a través d'una xarxa neuronal feedforward (FFN) amb conexions residuals i capes de normalització.

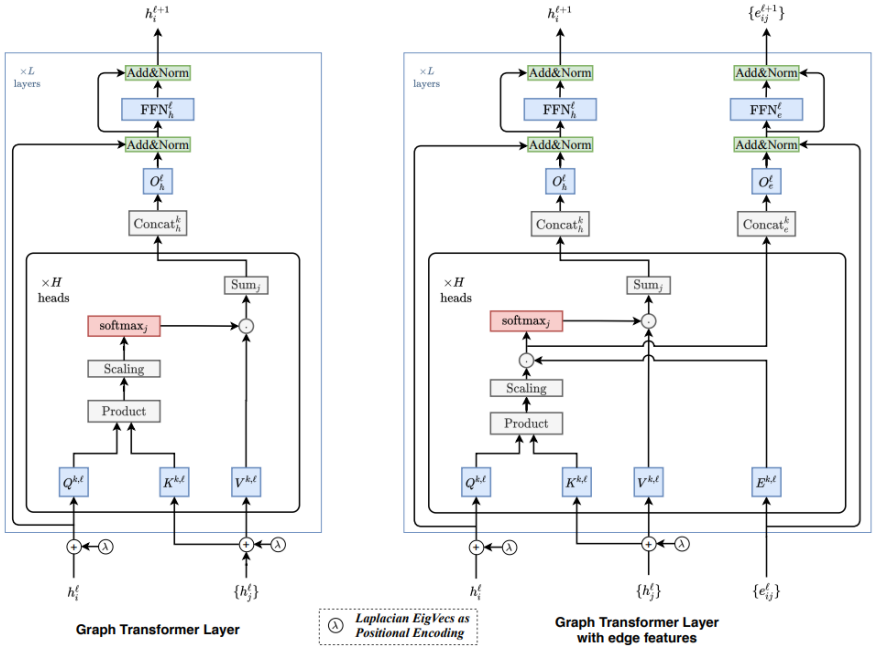


Figura 4.Arquitectura de GT i GT(E)

Capa de GT amb característiques d'arcs.

La capa de GT amb característiques d'arcs està dissenyada per utilitzar millor la informació de característiques riques disponibles en varis conjunts de dades de grafs en forma d'atributs d'arcs. La idea és aprofitar la informació disponible en als arcs, que són puntuacions de pars corresponents a una parell de nodes, i millorar les puntuacions d'atenció implicats computadors mitjançant "Implicit Attention". Això s'aconsegueix multiplicant les puntuacions d'"Implicit Attention" amb les característiques dels arcs disponibles, el que permet injectar aquesta informació addicional en el procés d'"Attention".

Les equacions d'actualització de capa per a la capa *l* són similar s a les capes de GT estàndard, però s'agreguen terminis addicionals que tenen en conta les característiques dels arcs:

$$
\begin{aligned}
\hat{h}_i^{\ell+1} &= O_h^{\ell} \Big\|_{k=1}^{H} \Big(\sum_{j \in \mathcal{N}_i} w_{ij}^{k,\ell} V^{k,\ell} h_j^{\ell} \Big), \\
\hat{e}_{ij}^{\ell+1} &= O_e^{\ell} \Big\|_{k=1}^{H} \Big(\hat{w}_{ij}^{k,\ell} \Big), \quad \text{where,} \\
w_{ij}^{k,\ell} &= \text{softmax}_j(\hat{w}_{ij}^{k,\ell}), \\
\hat{w}_{ij}^{k,\ell} &= \Big(\frac{Q^{k,\ell} h_i^{\ell} \cdot K^{k,\ell} h_j^{\ell}}{\sqrt{d_k}} \Big) \cdot E^{k,\ell} e_{ij}^{\ell},
\end{aligned}
\qquad \textbf{(5)}
$$

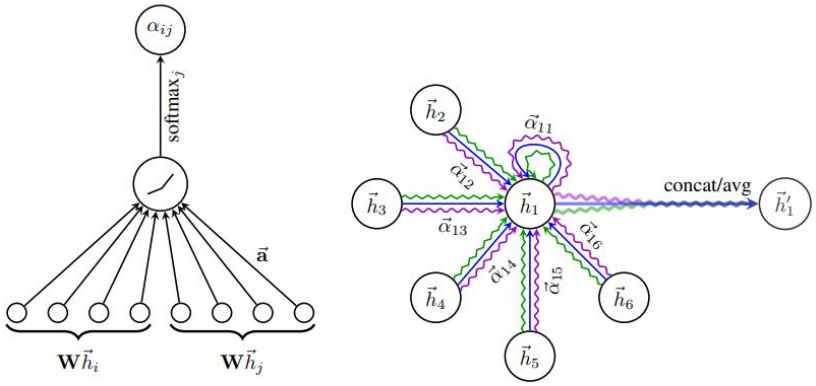
On e_{ij}^{l+1} és la representació de característiques dels arcs *(i, j)*, i $E^{k,l}$ és una matriu de paràemtres per transformar aquestes característiques. Les representacions actualizades dels nodes i arcs es passen a través de FFNs separats amb conexións residuals i capes de normalització.

3.2.2 GAT

Els Graph Attention Networks (GAT) milloren el processament de dades en grafs mitjançant un mecanisme que permet a cada node considerar la importància dels seus veïns de forma adaptativa (Velickovi, 2018). Aquest enfocament comença amb l'aplicació d'una transformació lineal a les característiques de cada node, utilitzant una matriu de pesos *W* que és apresa durant l'entrenament. Per a cada par de nodes connectats, *i* i *j*, es calcula un coeficient d'atenció preliminar utilitzant una combinació de les característiques transformades d'ambdós nodes. Aquest càlcul es realitza de la següent manera:

$$e_{ij} = LeakyReLU(a^T[Wh_i||Wh_j]) \quad \textbf{(6)}$$

on || denota l'operació de concatenació i *a* és un vector de paràmetres també après. I *LeakyReLU* és una funció d'activació que permet certa propagació de valors negatius per a mantenir gradients en l'entrenament de la xarxa.



Una vegada obtinguts els coeficients preliminars e_{ij}, es normalitza utilitzant la funció "softmax" per a convertir-los en una distribució de probabilitats que modela la importància relativa de cada node veí:

$$a_{ij} = \frac{exp(e_{ij})}{\sum k \epsilon N_i exp(e_{ik})} \quad \textbf{(7)}$$

on N_i és el conjunt de tots els veïns del node *i*, fins i tot el propi *i* si el graf permet llaços.

Després, el model actualitza les característiques de cada node amb una combinació ponderada de les característiques dels seus veïns, escalats pels seus coeficients d'atenció normalitzats, la qual cosa pot expressar-se com:

$$h_i^l = \sigma(\Sigma j \epsilon N_i \alpha_{ij} W h_j) \quad \textbf{(8)}$$

on σ és una funció d'activació no lineal, comunament ReLU o sigmoidal.

Per a capturar múltiples aspectes de la informació i millorar la capacitat del model per aprendre patrons complexos, els GAT acostumen a utilitzar el que es denomina "multi-head attention". En aquest enfocament varies instancies independentment del mecanisme d'atenció descrit anteriorment s'executa en paral·lel. Les sortides de cada "head" d'atenció poden ser concatenades o mitjanes, cosa que proporciona una representació més rica i estable de les característiques dels nodes.

Aquest mecanisme d'atenció "multi-head" no només refia la capacitat del model per a discernir i processar diferents tipus d'informació relacional en el graf, sinó que també ajuda a estabilitzar l'aprenentatge en diversificar les vistes de les característiques del node que el model està considerant.

3.2.3 GIN

Per a modelar l'agregació de veïns en GIN, s'utilitza funcions multiconjunt profunda. Aquestes funcions capturen l'estructura del graf d'entrada i garanteix la injectivitat en l'agregació de veïns (Xu, 2019). Denotem X com un multiconjunt de nodes veïns d'un node v, i $f : X \rightarrow \mathbb{R}^n$ com una funció que assigna cada node en X a un vector de característiques $\mathbb{R}^n$.

L'actualització dels nodes en GIN es realitzen utilitzant una combinació de la funció f i una funció de suma. Sigui $h_v^{(k)}$ la representació del node v en la capa k, i $N(v)$ el conjunt de nodes veïns de v.

L'actualització d'un node v en la capa k es realitza de la següent manera:

$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot h_v^{(k-1)} + \sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}\right) \quad \textbf{(9)}$$

On:

- $MLP^{(k)}$ és un perceptró multicapa que modela la funció f per a la capa k.
- $h_v^{(k-1)}$ és la representació del node v en la capa anterior.
- $h_u^{(k-1)}$ és la representació del node veí u en la capa anterior.

Gin utilitza funcions multiconjunt profunda i la funció suma per actualitzar les representacions dels node en cada capa de la xarxa. Aquesta arquitectura garanteix la injectivitat en l'agregació de veïns i permet captura eficaçment la informació estructural en el grafs d'entrada.

3.2.4 GCN

Les xarxes convolucionals de grafs (GCN) ofereixen un enfocament sofisticat per a l'aprenentatge de dades estructurades en grafs, especialment útil en escenaris d'aprenentatge semisupervisat on només un subconjunt de nodes té etiquetes (Kipf, 2017). La idea bàsica darrere dels GCN és entendre les potents xarxes neuronals convolucionals (CNN) des de quadrícules regulars fins a estructures de gràfics irregulars.

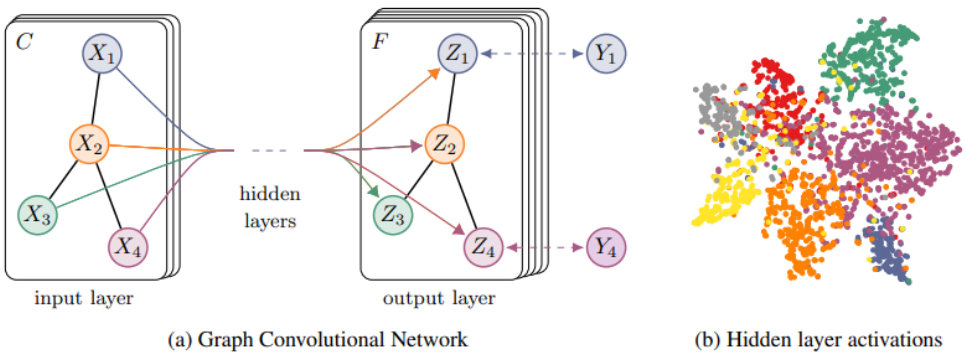


Figura 5. Representació esquemàtica del GCN amb canals d'entrada C i mapes de característiques F a la capa de sortida

Els GCN funcionen aprofitant la teoria espectral dels grafs d'espais euclidians fins a dades estructurades per grafs. Cada capa d'un GCN calcula noves representacions de nodes agregant les característiques dels nodes veïns, suavitzant eficaçment les característiques del graf alhora que captura estructures de grafs locals i globals.

1. Cada node ***i*** del graf està representat per un vector de característiques x_i. Tot el graf està descrit per una matriu d'adjacència *A* i una matriu de graus *D*, on D_{ii} és la suma dels pesos de totes les arestes connectades al node *i*.
2. Abans d'agregar les característiques veïnes, les característiques del node es transformen mitjançant una matriu de pes *W*, que es comparteix entre tots els nodes. Aquesta transformació alinea les dimensions de les característiques del node amb la sortida desitjada i introdueix paràmetres entrenables al model.
3. Per propagar característiques de manera efectiva a través del graf, el GCN utilitzen la matriu d'adjacència *A* del graf, combinada amb la matriu de graus *D*. Aquest pas és crucial per comprendre com flueix la informació a través de la xarxa.
4. Per evitar que l'escala dels vectors de característiques creixi molt i garantir que les característiques de diferents parts del graf contribueixin per igual, la matriu d'adjacència sovint es normalitza. La tècnica de normalització simètrica utilitzada implica calcular $D^{-1/2} AD^{-1/2}$. Aquesta normalització ajuda a mantenir una influència equilibrada dels veïns de cada node.
5. El funcionament fonamental dels GCN es pot descriure mitjançant la regla d'actualitzacions per capes:

$$H^{(l+1)} = \sigma(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^{(l)} W^{(l)}) \quad \textbf{(10)}$$

   Aquí, H^l representa la matriu de característiques del node a la capa *l*, W^l és la matriu de pes a la ***l*** capa, i *sigma* denota una funció d'activació no lineal com ReLU. Aquesta regla vol dir que les noves característiques de cada node són un agregat a les seves pròpies característiques i les dels seus veïns, transformades per W^l**,** i processades mitjançant una funció d'activació.
6. En apilar diverses capes d'aquestes operacions, els GCN poden capturar informació de veïnat d'orde superior. Cada capa agrega característiques d'un veïnat cada cop major al graf, cosa que permet a la xarxa aprendre patrons complexos al llarg de múltiples salts entre nodes.
7. Igual que en les xarxes neuronals tradicionals, els paràmetres (les matrius de pesos W^l) s'aprenen minimitzant una funció de pèrdua mitjançant retropropagació. La naturalesa semisupervisada de les dades significa que la pèrdua només es calcula per als nodes etiquetats, però els gradients es propaguen per tot el graf permetent que els nodes no etiquetats aprenguin dels seus veïns etiquetats.

3.2.5 MPNN

Les xarxes neuronals de pas de missatges (MPNN) orientat a grafs utilitzen un enfocament sistemàtic per processar la informació en estructures on les entitats i les seves relacions en modelen com a nodes i arcs, respectivament. (Justin Gilmer, 2017)El procés consta de 2 fases principals: pas de missatge i lectura:

1. Durant aquesta fase, cada node agafa informació dels seus veïns immediats a través d'una funció de missatge definida:

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \quad \textbf{(11)}$$

on $M^{t+1}{}_v$ és el missatge rebut pel node ***v*** en el pas *t+1*, *N(v)* representa els veïns *v*, $h^t{}_v$ i $h^t{}_w$ són les característiques del node veí y el node actual, respectivament. I e_{vw} denota les característiques de l'arc entre els nodes *w* i *v*. La funció *M* encapsula la lògica per combinar aquestes característiques en un missatge.

Una vegada generats els missatges, cada node actualitza el seu estat basant-se en el seu estat actual i els missatges entrants. Aquesta actualització es captura mitjançant:

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \quad \textbf{(12)}$$

Aquí, *U* és la funció d'actualització que transforma l'estat actual del node $h^t{}_v$ integrant el nou missatge $m^{t+1}{}_v$.

2. Després de diverses iteracions de transmissió de missatges, la funció de lectura agrega les característiques de tots els nodes per obtenir un descriptor o una sortida de graf global, que s'utilitza sovint per a tasques com la classificació de grafs o a la regressió. Aquesta funció d'agregació denotada com a *R*, combina tots els estats del node de la iteració final per produir la sortida:

$$\hat{y} = R(\{h_v^T \mid v \in G\}) \quad \textbf{(13)}$$

On *r* és la sortida final, *T* indica l'últim pas de temps i *G* és el graf en qüestió.

4 MolGraph

MolGraph és una llibreria de Python dissenyada específicament per integrar la tecnologia de xarxes neuronals gràfiques (GNNs) amb l'anàlisi d'estructures moleculars, facilitant l'aplicació de tècniques avançades d'aprenentatge profund per a predir i explorar propietats moleculars. Aquesta biblioteca és destacada per la seva estreta integració amb TensorFlow i Keras, la qual cosa permet als usuaris dissenyar i entrenar models de GNN utilitzant la API de Keras, coneguda per la seva facilitat d'ús i flexibilitat. Aquesta compatibilitat assegura que els usuaris puguin aprofitar les capacitats avançades d'autodiferenciació i escalabilitat amb diverses arquitectures d'una manera accessible. (Alexander Kensert, 2023)

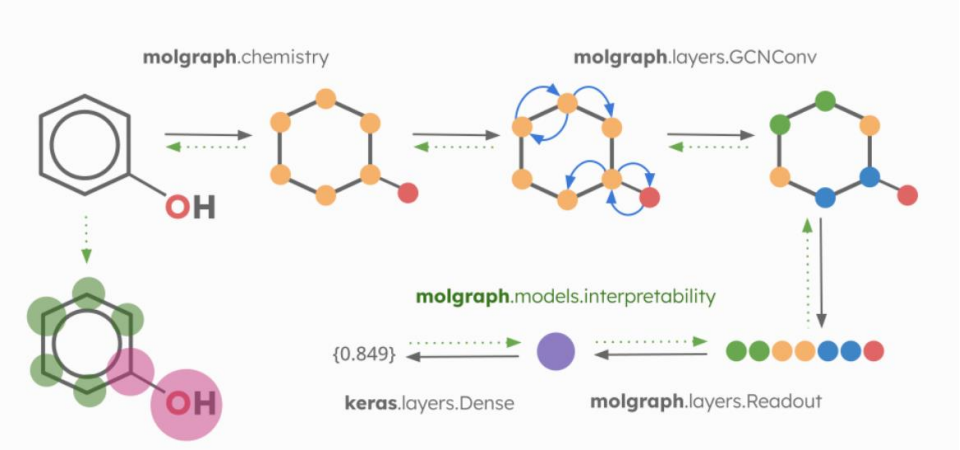


Figura 6. Esquema funcionament de la llibreria MolGraph

MolGraph s'especialitza en el maneig de grafs moleculars, proporcionant funcionalitats específiques per convertir estructures químiques com les representades en SMILES o arxius .mol, en grafs on els nodes i les arestes representen àtoms i enllaços químics, respectivament. Això inclou la creació de grafs que capturen amb precisió l'estructura i la connectivitat molecular, essencial per a la modelització precisa de les interaccions i les propietats químiques.

La llibreria inclou una àmplia varietat de capes de GNN dissenyades específicament per treballar amb dades estructurades en format de graf. Aquestes capes, com Graph Convolutional Network (GCN) i Graph Attention Networks (GAT), permeten als usuaris capturar diferents aspectes de la informació estructural i química continguda en els grafs. Cada tipus de capa ofereix una perspectiva única per modelar les relacions entre àtoms i enllaços, proporcionant una eina poderosa per analitzar i predir propietats moleculars.

MolGraph també facilita el preprocessament i la codificació de grafs moleculars, incloent-hi la generació de característiques per a node i arestes basat en propietats atòmiques i enllaços. Aquestes característiques son essencials per a l'entrenament efectiu dels modes de GNN, i MolGraph proporciona eines robustes per a la seva normalització i optimització, assegurant que les dades estiguin ben preparades per a l'aprenentatge automàtic.

Dissenyada per ser accessible i fàcil d'utilitzar, MolGraph és dirigit tant a científics sense una profunda experiència en aprenentatge automàtic com a desenvolupadors. A més, els usuaris poden personalitzar la llibreria afegint noves capes GNN, modificant les existents, o integrant MolGraph amb altres eines i llibreries per ampliar encara més la seva funcionalitat.

5 Base de Dades QM7

La base de dades QM7 és essencial per als estudis de química computacional i aprenentatge automàtic en la llibreria MolGraph, proporcionant un recurs valuós per al desenvolupament i validació de models predictius. Aquesta base de dades està formada per aproximadament 6449 molècules orgàniques, que inclouen fins a 23 àtoms d'hidrogen, carboni, oxigen i nitrogen, representant una àmplia gamma de compostos orgànics. La seva utilitat en l'avaluació de mètodes d'aprenentatge automàtic és degut a la precisió de les seves dades, derivades de càlculs de la teoria del funcional de la densitat (DFT), que inclouen propietats fonamentals com l'energia total atòmica, les energies de les òrbites moleculars més alta ocupada (HOMO) i més baix desocupat (LUMO), i la diferència energètica entre aquests orbitals, indicadors clau de la reactivitat química i l'estabilitat molecular.

Les estructures moleculars dins de QM7 es representen en format .sdf (Structure Data File), un format comunament utilitzat per descriure estructures moleculars i les seves propietats químiques associades en una forma estàndard i llegible per màquina. El format .sdf permet emmagatzemar molècules, on cada molècula està acompanyada d'un conjunt de metadades i valor de propietats que poden incloure no només la configuració espacial dels àtoms, sinó, també propietats calculades com energies d'orbital i altres característiques químiques rellevants.

SDF

```
gdb7k_0000.xyz
 OpenBabel11221611003D

  5  2  0  0  0  0  0  0  0  0  0999 V2000
    0.9778   -0.0025   -0.0044 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.0953   -0.0024    0.0041 H   0  0  0  0  0  0  0  0  0  0  0  0
    3.7268    1.0269    0.0041 H   0  0  0  0  0  0  0  0  0  0  0  0
    1.6782   -0.5277    0.8781 H   0  0  0  0  0  0  0  0  0  0  0  0
   -3.5973   -0.5075   -0.9054 H   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  1  4  1  0  0  0  0
M  RAD  3   1   3   3   2   5   2
M  END
```

Figura 7.Representació d'una molècula en format .sdf

Cada línia i número té una finalitat per descriure completament l'estructura molecular (SDF file, s.f.):

1. **'gdb7k_0000.xyz'**: Identifica el nom de l'arxiu o l'etiqueta de la molècula.
2. **'OpenBabel11221611030D'**: Mostra que l'arxiu va ser processar o generat per OpenBable, que és una eina de software per a la conversió de formats d'arxius químics.
3. **'5 2 0 0 0 0 0 0 0 0999 V2000'**: El número **'5'** indica el nombre d'àtoms i el número **'2'** indica el nombre d'enllaços. Els següents valors **'0'** representen comptadors per a altres elements com anells, que no són rellevants en aquest cas. I **'999 V2000'** és un indicador de format per arxius .sdf, indicant que aquest arxiu segueix l'estàndard V2000
4. Les següents línies abans de les connexions d'enllaç descriuen cada àtom.

‘0.9978 -0.0025 -0.0044 C 0 0 0 0 0 0’: els 3 primers valors indiquen les coordenades cartesianes de l’àtom X, Y, Z. ‘C’ és el símbol de l’element, carboni. I els números següents ‘0’ són reservats per a informació addicional sobre l’àtom.

5. **‘1 2 1 0 0 0 0’**: Indica un enllaç entre l’àtom 1 i el l’àtom 2. El següent **‘1’** indica el tipus de l’enllaç (1=simple, 2=doble, etc). I els ‘0’ addicionals són per informació d’estereoquímica i altres detalls específics de l’enllaç.
6. **‘M RAD 3 1 3 3 2 5 2’** : Representa informació de radicals o carregues en els àtoms, on els números indiquen l’àtom i el tipus de radical o càrrega.

5.1 Metodologia d’Execució de Models

En aquest estudi, s’emplena una metodologia sistemàtica per al desenvolupament i l’optimització de models de xarxes neuronals gràfiques (GNN) utilitzant la base de dades QM7. L’objectiu és explorar diverses configuracions de models per a determinar les més efectives en termes de minimitzar l’error quadràtic mitjà (MSE).

5.1.1 Configuració i proves dels Models

Es van dur a terme experiments amb diferents tipus d’arquitectures GNN, incloses Graph Transformer (GT), Graph Isomorphism Network (GIN), Graph Convolutional Network (GCNN), Gated Graph Convolutional Network (GatedGCN) i Graph Attention Network (GAT). Per a cada tipus de model, es va avaluar diferents configuracions:

Graph Transformer (GT):

Es va experimentar amb diverses estructures, variant des de 3 capes GT amb 512 nodes fins a 4 capes GT amb 1024 nodes, ajustant les capes denses i la tassa d’aprenentatge (lr) entre 0.0001 i 0.01. El nombre d’èpoques d’entrenament varia entre 100 i 200, depenent de la configuració específica:

GT	Dense	LR	Epochs
1->1024 nodes, 3->512 nodes	3->1024 nodes	1-e3	200
1->1024 nodes, 1->512 nodes	3->1024 nodes	1-e2	200
1->1024 nodes	3->1024 nodes	1-e2	200
3->1024 nodes	----------	1-e2	200

Taula 1. Arquitectura del model GT que s'han testejat

Graph Isomorphism Network (GIN):

Les proves inclouen configuracions des de 3 capes Gin de 1024 nodes fins a configuracions de 512 nodes, totes utilitzant un “lr” de 0.01 i entrenades durant 200 èpoques:

GIN	Dense	LR	Epochs
3->1024 nodes	1->1024 nodes	1e-2	200
1->1024 nodes	----------	1e-2	200
3->512 nodes	----------	1e-2	200
2->1024 nodes	----------	1e-2	200

Taula 2. Arquitectura del model GIN que s'han testejat

Altres arquitectures GNN (GCN, GatedGCN, GAT):

Amb els altres models es va aplicar la mateixa estratègia degut als resultats observats anteriorment amb els altres models. Es va aplicar models amb 3 capes de 1024 nodes i una capa densa amb la funció d'activació ReLU. Utilitzant un "lr" d'1e-2 i un cicle d'entrenament de 200 èpoques per a totes les proves.

Cada model està rigorosament entrenat i avaluat utilitzant el MSE obtingut en les proves i comprovacions per determinar l'efectivitat de les configuracions. Es van anar realitzant ajustos iteratius en les configuracions dels models per millorar contínuament el seu rendiment. A més, aquesta metodologia detallada assegura una avaluació exhaustiva dels models, identificant les configuracions òptimes de manera precisa.

6 Base de dades URV

La base de dades de la URV està formada per un directori que conté 344 fitxers .sdf. Cada fitxer representa una molècula, i està estructurada de la següent manera:

- **Encapçalat:** Les dues primeres línies típicament inclouen el nom o identificador de la molècula i comentaries addicionals que es refereixen al programari utilitzat (OEChem) per generar el fitxer.
- **Comptatge d'àtoms i enllaços:** Una línia inicial indica el nombre d'àtoms i enllaços en la molècula, seguit per detalls específics que poden incloure versions de format (V2000).
- **Descripció d'àtoms:** Seqüències de línies descriuen cada àtom en la molècula, detallant les seves coordenades tridimensionals (X, Y, Z), l'element químic, i altres marcadors d'estereoquímica o estats electrònics que no s'utilitzen comunament en descripcions simples.
- **Descripció d'enllaços:** Línies subsegüents detallen els enllaços entre par d'àtoms, especificant quins àtoms estan connectats i el tipus d'enllaç (simple, doble, etc)
- **Metadades i propietats:** Al final de l'arxiu es poden incloure propietats calculades com SMILES (una notació simplificada que representa l'estructura de les molècules), InChi y InChiKey (identificadors que proporcionen una representació textual única de la substància) i la fórmula molecular.

També hi ha un fitxer que indica l'afinitat de cada molècula. En bioquímica i farmacologia, l' "**Affinity**" és la força de la interacció entre un lligand (la nostra molècula) i una proteïna, com un receptor o enzim. És una mesura crucial per avaluar l'eficiència potencial d'una molècula com a fàrmac, ja que una major afinitat per al receptor objectiu generalment correlaciona amb una major activitat biològica.

6.1 Metodologia d'Execució de Models

L'execució dels models i programes dissenyats per treballar amb la base de dades de la URV es duu a terme mitjançant un procés detalladament planificat i automatitzat. Aquest procés s'implementa amb la finalitat d'avaluar l'eficàcia de diversos models de GNNs en la predicció de propietats moleculars, utilitzant múltiples conjunts de dades d'entrenament per garantir la robustesa i fiabilitat dels resultats en un context de disponibilitat limitada de dades.

6.1.1 Preparació de dades

Donada la limitada quantitat de dades disponibles en la base de dades de la URV, s'ha optat per dividir el conjunt de dades s'ha dividit de forma aleatòria en tres grups distints d'entrenament i test. Aquesta estratègia permet avaluar la consistència dels models en diferents mostres de dades i minimitzar les possibilitats que els resultats obtinguts siguin producte de l'atzar. Per a fer-ho s'ha barrejat el conjunt de dades de manera aleatòria utilitzant 3 llavors diferents (llavor=1, llavor=2, llavor=3). Després, s'han seleccionat els primers 75% dels elements com a dades d'entrenament i els últims 25% com a dades de prova.

6.1.2 Configuració i entrenament de Models

Els models de GNN seleccionats per a aquest estudi es descriuen a continuació, cada un amb una configuració específica de capes, números d'èpoques i taxa d'aprenentatge:

1. Model de Xarxa Convolucional(GCN)
    a. **Capes**: "GCN,1024,;GCN,256,;GCN,256,;Dense,256,relu;Dense,1,"
    b. **Nombre d'èpoques**: 30
    c. **Ràtio d'aprenentatge**: 1e-3
2. Model de Transformer de Grafs(GT)
    a. **Capes**: "GT,128,;GT,64,;GT,64,;Dense,64,relu;Dense,1,"
    b. **Nombre d'èpoques**: 30
    c. **Ràtio d'aprenentatge**: 1e-3
3. Model de Xarxa d'Atenció de Grafs(GAT)
    a. **Capes**: "GAT,128,;GAT,64,;GAT,64,;Dense,64,relu;Dense,1,"
    b. **Nombre d'èpoques**: 30
    c. **Ràtio d'aprenentatge**: 1e-3
4. Model de Xarxa Isomòrfica de Grafs(GIN)
    a. **Capes**: "GIN,128,;GIN,64,;GIN,64,;Dense,64,relu;Dense,1,"
    b. **Nombre d'èpoques**: 30
    c. **Ràtio d'aprenentatge**: 1e-3
5. Model de GCN amb comportes(GatedGCN)
    a. **Capes**: "GatedGCN,128,;GatedGCN,64,;GatedGCN,64,;Dense,64,relu;Dense,1,"
    b. **Nombre d'èpoques**: 30
    c. **Ràtio d'aprenentatge**: 1e-3

Cada model és entrenat utilitzant el seu respectiu grup de dades d'entrenament i posteriorment avaluat amb el conjunt de proves corresponent. Els detalls específics de configuració del model asseguren que cada model està optimitzat per al tipus d'estructura de dades i les característiques de la base de dades de la URV.

7 Models Predictius amb la Base de Dades de la URV

7.1 Descripció i funcionament dels programes desenvolupats

Els programes estan dissenyats per aquest projecte però facilitar la implementació, avaluació i ajust de models predictius utilitzant la base de dades de la URV, adaptats per a operar amb la biblioteca MolGraph. Els programes estan desenvolupats en Python i estan estructurats per a ser utilitzats en tres fases consecutives: traducció de la base de dades de la universitat al format que requereix Molgraph, creació i entrenament del model, i testatge del model entrenat.

Els tres programes estan dissenyats per ser utilitzat en seqüència i estan integrats a través d'una interfície de línia de comandes. A més, s'ha establert una documentació detallada per a cada programa, garantint que els usuaris puguin entendre i operar cada aspecte del procés.

A continuació es detalla cada un d'aquests programes.

7.1.1 Programa 1: Traducció de la base de dades de la URV

Aquest programa és essencial per a l'adequada manipulació i preparació de les dades moleculars per al seu és en models predictius desenvolupats amb MolGraph. Aquest programa realitza vàries funcions crítiques: llegeix i processa arxiu d'estructura molecular en format SDF, els combina amb les dades d'afinitat molecular, i genera conjunt de dades d'entrenament i prova. A més, genera un informe detallat sobre qualsevol discrepància o error durant el procés. A continuació es detalla el funcionament del programa:

7.1.1.1 Lectura de Dades Moleculars i Afinitat

El programa comença per la lectura de l'arxiu SDF dins d'una carpeta específica anomenada "Ligand". Aquests arxius contenen la representació estructural de les molècules. Cada arxiu es llegeix completament com una cadena de text, processant i eliminant qualsevol marcador final innecessari per a assegurar la neteja de les dades.

Paral·lelament, es llegeix un arxiu de text que llista les afinitats de diverses molècules, identificades per un ID. Aquest arxiu és crucial per a relacionar cada estructura molecular amb la seva respectiva afinitat, un factor clau per a l'entrenament de models predictius.

7.1.1.2 Fusió i validació de Dades

Les estructures moleculars i les dades d'afinitat es combinen mitjançant els seus IDs corresponents. El programa utilitza operacions de fusió de pandes per a integrar ambdues fonts de dades, assegurant que cada molècula tingui la seva afinitat corresponent.

A continuació es realitza una verificació per a identificar i reportar molècules que apareixen únicament en els arxius SDF o només en l'arxiu d'afinitat. Això és crucial per a garantir la integritat de les dades i per a entendre les possibles fonts d'error o discrepància en la base de dades.

7.1.1.3 Divisió en Conjunts d'Entrenament i Prova

Utilitzant un mètode de permutació aleatòria, el conjunt de dades es divideix en subconjunt per a entrenament i prova basat en un percentatge definit. Això permet avaluar la capacitat predictiva del model en dades no vistes durant l'entrenament.

Els subconjunts resultants s'ajunten perquè contenguin només les columnes necessàries, típicament 'Molecule' i 'Affinity', anomenades per facilitar el seu ús en entrenaments futurs.

7.1.1.4 Report de resultats i errors

Finalment, es crea un arxiu de sortida que documenta el nombre de files i la proporció de dades en cada subconjunt d'entrenament i prova. A més, s'inclou una mostra de les dades per verificar visualment el seu format i exactitud. El report també destaca qualsevol inconsistència encontrada duran el procés de fusió de les dades.

7.1.2 Programa 2: Creació i entrenament del model

El programa permet la construcció de models d'aprenentatge profund a través d'una cadena de text, que detalla les capes que compondran el model. Aquesta cadena té un format específic on cada element de la capa se separa per comes(','), i cada capa està delimitada amb un punt i coma(';'). Els elements de cada capa inclou el tipus de capa, el nombre de nodes i, opcionalment, la funció d'activació. En aquest disseny possibilita la configuració de models amb una gran varietat d'arquitectures de xarxa.

7.1.2.1 Exemple d'interpretació de la Cadena

Format General: 'TipusCapa, nodes, FunciócActivació'

Cadena d'exemple: 'Dense,10,relu;Dense,20,softmax'

Aquí, el model es configura inicialment amb una capa densa('Dense') de 10 nodes utilitzant la funció d'activació ReLU, seguida per una altra capa densa de 20 nodes utilitzant softmax com a funció d'activació.

7.1.2.2 Integració i personalització de Capes

L'arxiu de configuració 'layers.py' conté un diccionari que associa nombres de capes amb les seves respectives implementacions en TensorFlow o Molgraph. Aquest arxiu es clau per a l'expansió i adaptació del conjunt de capes disponibles sense modificar el codi base del programa.

L'arxiu defineix un mapa('type_layer_class') on les claus són nombres de capes com strings, i els valors són referències a classes de capes específiques en TensorFlow o MolGraph.

Els usuaris poden afegir noves capes o modificar les existents simplement editant aquest arxiu, la qual cosa permet adaptar ràpidament el programa a nous avanços en tècniques de modelatge o a requisits específics de projectes particulars.

7.1.2.3 Codificació de Molécules

El tractament adequat de les estructures moleculars es fonamental per a l'entrenament efectiu de models en química computacional. El programa utilitza codificadors que poden transformar aquestes estructures en formats adequats per al processament de xarxes neuronals.

L'arxiu de configuració 'encoder.py' conté els diferents codificadors com 'encoder_3d' i 'encoder_2d' que converteixen les estructures en representacions vectorials en 3D o 2D respectivament. A més, a l'estar definits en un arxiu separat, els codificadors poden ser modificats o estesos per a implementar noves tècniques de codificació molecular sense alterar la resta del programa.

7.1.2.4 Flux d'Entrenament del Model

El procés d'entrenament es realitza en vaires etapes consecutives que asseguren l'adequada construcció, configuració, execució i avaluació del model.

Abans de l'entrenament, les dades moleculars són transformades pels codificadors seleccionat, preparant els "datasets" d'entrenaments. Després, la cadena d'especificacions del model es llegeix i es processa per construir l'arquitectura desitjada. Durant aquest procés, s'inicialitza les capes segons el tipus i els paràmetres especificats en la cadena, utilitzant l'arxiu de capes per a resoldre les referències a classes de capes.

Una vegada tenim el model, es compila amb un optimitzador de Adam i amb una funció de pèrdua específica, com l'error absolut mitjà. (MAE) S'executa l'entrenament del model utilitzant les dades d'entrada preparades, optimitzant els pesos de la xarxa durant un número definits per l'usuari d'èpoques i es va registrant les mètriques de rendiment.

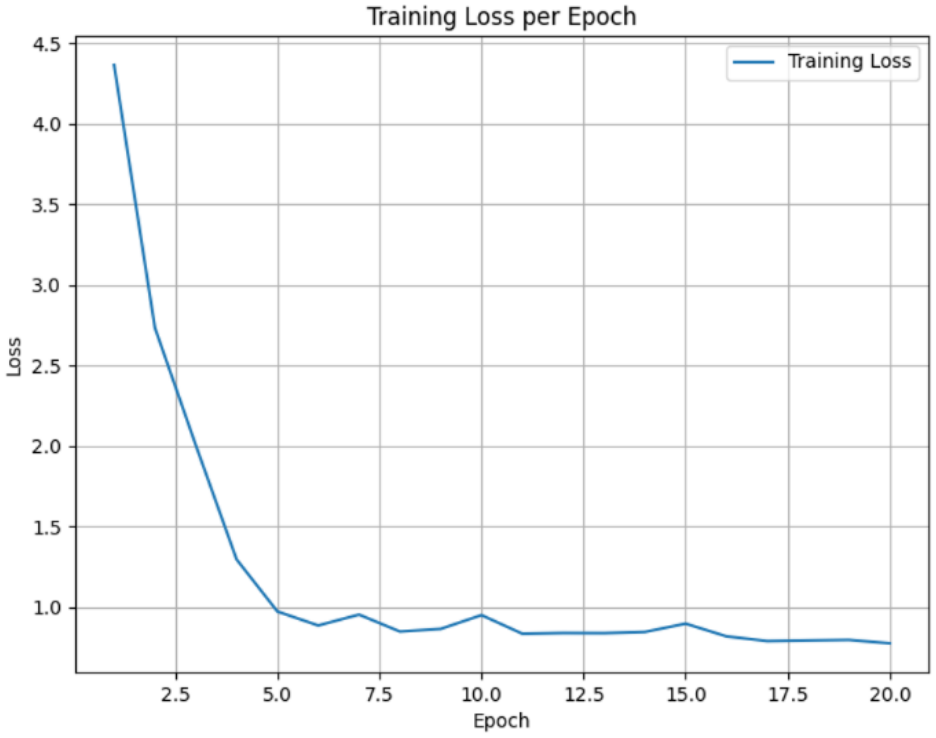


Figura 8. Gràfica que representa el MSE que obté el model GT amb base de dades de URV en cada "epoch"

Posteriorment, es genera una gràfica que mostra l'evolució de la pèrdua d'entrenament al llarg de les èpoques, facilitant l'avaluació visual de l'aprenentatge del model. A més, el model entrenat s'emmagatzema per al seu ús en un futur, i es genera un report detallat sobre els resultats i les configuracions utilitzades en l'entrenament.

Aquest flux detallat i estructura del programa asseguren que els usuaris puguin configurar, entrenar i avaluar models predictius de manera eficient i adaptables, facilitant la investigació i el desenvolupament en el camp de la química computacional i altres àrees relacionades.

7.1.3 Programa 3: Testegi del model entrenat

Aquest component del sistema és fonamental per validar l'eficàcia del model predictiu una vegada entrenada. El programa agafa un conjunt de dades de proves i el model entrenat que indica l'usuari, executant una sèrie d'operacions que permeten avaluar

l'exercici del model mitjançant mètriques d'errors i visualitzacions comparatives entre valors predits i reals.

7.1.3.1 Codificació de Dades de Prova

El procés de testaci comença amb la preparació de les dades de prova. Depenent de si la codificació de les molècules és en 3D o 2D, se selecciona el codificador adequat ("encoder_3d" o "encoder_2d"). Aquesta selecció es basa en el paràmetre "encoder3D" proporcionat, on s'utilitza "encoder_3d" per transformar les estructures moleculars en representacions tridimensionals. I s'utilitza "encoder_2d" per convertir les estructures en representacions bidimensionals.

Aquests codificadors transformen les dades moleculars del conjunt de proves en un format numèric que el model pot processar efectivament.

7.1.3.2 Càrrega del model

El model prèviament entrenat es carrega utilitzant la funció "tf.keras.models.load_model". Aquest assegura que el model que està avaluant és exactament el que va ser optimitzat durant la fase d'entrenament, mantenint totes les configuracions i pesos.

7.1.3.3 Avaluació del model

Una vegada que les dades i el model estan preparats i carregats correctament, es procedeix a l'avaluació de la següent manera:

- El model, ara carregat, realitza prediccions sobre el conjunt de dades de prova. Aquestes prediccions són essencials per a comparar les respostes estimades del model amb els valors reals i observables en les dades de prova. Aquesta comparació és fonamental per avaluar com el model manejarà dades noves i no vistes, reflectint la seva capacitat de generalitzar en situacions pràctiques.
- Per quantificar el rendiment del model, s'utilitza la funció "model.evaluate()", que automàticament calcula mètriques de rendiment com l'error quadràtic mitja(MSE) i altres mètriques rellevant. Aquestes mètriques són crucials per a determinar l'eficiència del model i proporcionar una avaluació objectiva del seu exercici en les dades de prova.

La visualització és una eina clau per interpretar els resultats del testaci del model. Es realitzen les següents visualitzacions:

- Es genera una gràfica de dispersió que mostra els valors predits pel model en una comparació amb els valors reals. Aquesta visualització ajuda la precisió de les prediccions del model, mostrant gràficament que tan propers estan les prediccions als valors reals. Una bona alineació dels punts al llarg de la línia diagonal indica un model altament efectiu.

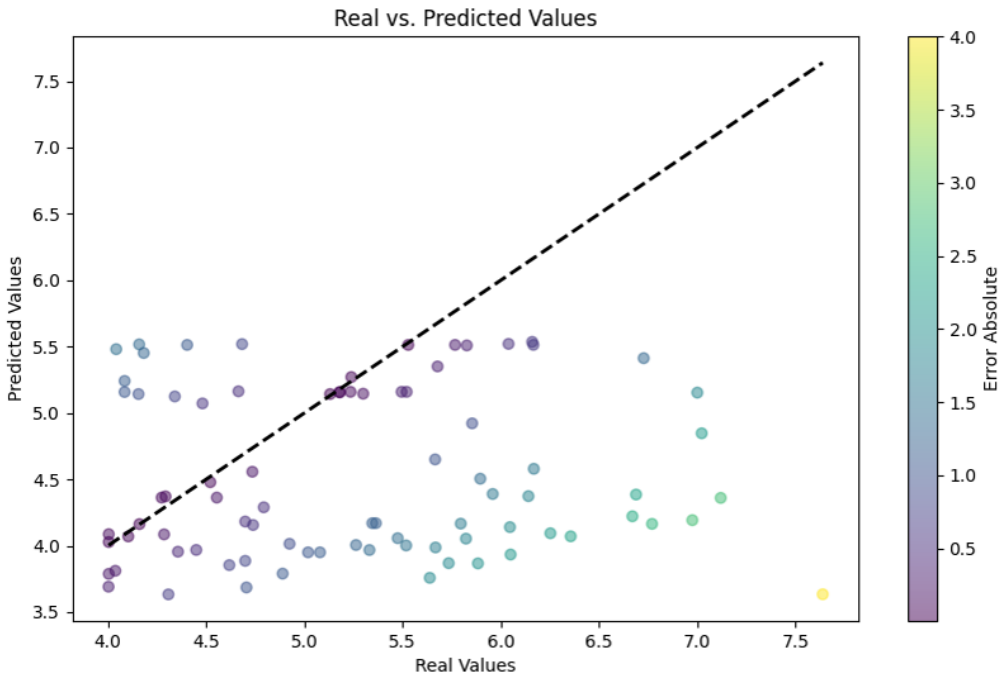


Figura 9. Gràfica que compara els valors predits pel model GT amb els valors real d'afinitat

- En la gràfica de dispersió, es traça una línia d'identitat (una línia diagonal) per a il·lustrar l'escenari ideal en el qual els valors predits coincideixen perfectament amb els valors reals. Aquesta línia serveix com a referència per avaluar l'exactitud de les prediccions.

Aquests passos d'avaluació i visualització permeten una comprensió detallada i clara del comportament i rendiment del model, assegurant que els usuaris i desenvolupadors tinguin la informació necessària per prendre decisions informades sobre l'ús o la millora del model predictiu.

8 Resultat i Discussió

8.1 Anàlisi dels resultats i optimització dels models amb QM7

Els resultats de les proves realitzades amb els models de xarxes neuronals gràfiques (GNN) utilitzant la base de dades QM7 es van analitzar i comparar amb els benchmarks documentats en l'estudi oficial de MolGraph. Aquesta anàlisi detallat ajuda a verificar la precisió i efectivitat de les nostres implementacions de models en la predicció de propietats moleculars.

	ELS MEUS MODELS	MOLGRAPH
GT	51.3474	7.5603
GIN	13.8631	19.2505
GCN	21.8294	18.9011
GATEDGCN	19.8819	10.2520
GAT	16.1843	17.9556
MPNN	55.2834	15.0094

Taula 3. Resultat MSE dels models GCN del document MolGraph i els creats per l'anàlisi

La taula mostra el MSE per a cada tipus de model GNN tant dels nostres experiments ("Els meus models") com el resultat reportat per MolGraph. Aquí es detalla l'anàlisi de cada model:

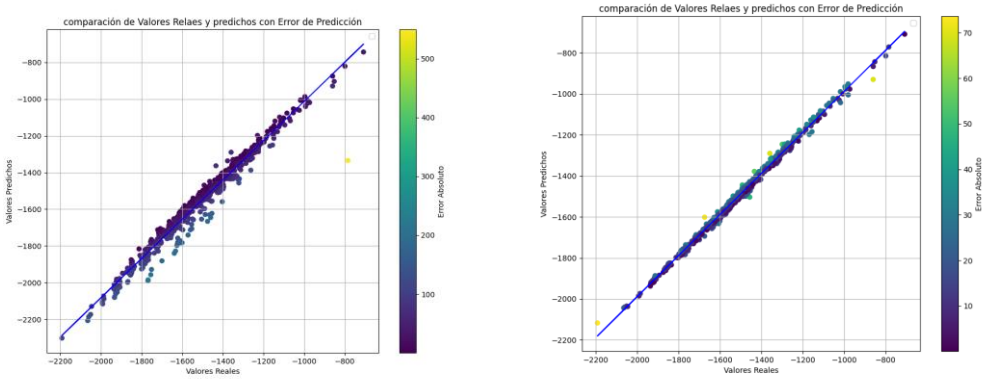


Figura 10. Utilitzant les dades QM7.Esquerra: Gràfica que compara els valors predits pel model GT amb els valors real. Dreta: Gràfica que compara els valors predits pel model GIN amb els valors real.

- **Graph Transformer:** Els nostres models GT mostren un MSE de 51.3474, que és significativament més alt que el MSE de 7.5403 reportat per MolGraph. Aquesta gran discrepància suggereix que és possible que hi hagi una diferència en la capacitat computacional empleada en els diferents estudis, donant resultats molt diferents.
- **Graph Isomorphism Network**: Amb un MSE de 13.8631 en el nostre model comparat amb 19.2505, s'observa que els nostres model Gin supera el rendiment del benchmark, destacant una implementació exitosa i possiblement una millor adaptació a les peculiaritats de la base de dades QM7.

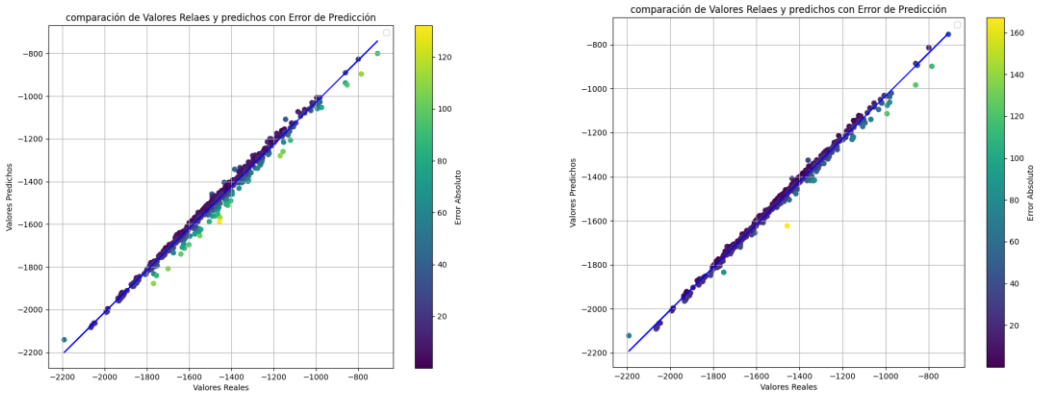


Figura 11 Utilitzant les dades QM7.Esquerra: Gràfica que compara els valors predits pel model GCN amb els valors real. Dreta: Gràfica que compara els valors predits pel model gatedGCN amb els valors real.

- **Graph Convolutional Network**: El MSE del nostre model GCN és 21.8294, lleugerament superior al del MolGraph de 18.9011, indicant un rendiment molt similar.
- **Gated Graph Convolutional Network:** El resultat de 19.8819 en comparació a 10.2520 de Molgraph mostra que encara que el nostre model funciona bé, encara hi ha una petita diferència.

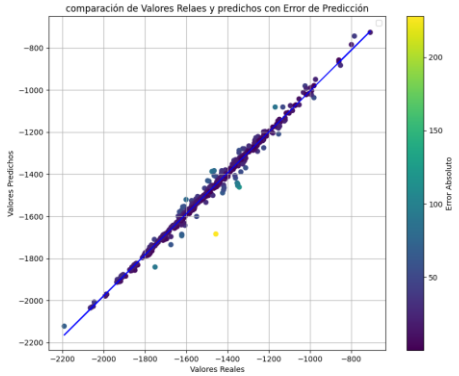


Figura 12. Utilitzant les dades QM7. Gràfica que compara els valors predits pel model GAT amb els valors reals

- **Graph Attention Network:** Amb un MSE de 16.1843 comparat amb 17.9556 de MolGraph, el nostre model mostra una petita millora respecte als resultats indicats pel document oficial. És a dir, tenen un rendiment semblant.

Amb aquesta anàlisi comparativa es demostra que, encara que la majoria dels nostres models mostren un rendiment semblant o superior al benchmark de MolGraph, alguns models, com el GT i el MPNN, requereix optimitzacions addicionals. La correlació generalment positiva entre els nostres resultats i els del benchmark confirma la validesa de les nostre implementacions de GNN, assegurant la confiança d'aplicar aquests models a altres conjunts de dades, com les de la URV, anticipant resultats igualment robustos i precisos.

8.2 Anàlisi dels resultats dels models amb la base de dades de la URV

Aqueta anàlisi aprofundeix en els resultats obtinguts de l'aplicació de diversos models de GNN a la base de dades de la URV. Per garantir la robustesa i la generalització dels models, el conjunt de dades ha estat dividit de tres maneres diferents, resultant en 3 grups de dades diferents sobre els quals s'entrenen els models.

	GRUP 1	GRUP 2	GRUP 3	TEMPS/EPOCH
GCN	2.8701	0.9455	0.7926	10s
GT	0.7840	0.8277	0.834	26s
GIN	2.6637	2.7103	2.6212	3s
GATEDGCN	3.1371	2.8123	3.6751	17s
GAT	3.7277	3.5489	3.6992	20s

Taula 4. Valors MSE obtinguts amb els models de GNN amb la base de dades de la URV

La taula proporcionada dona els errors quadràtics mitjans (MSE) obtinguts per cada model en els tres grups de dades, juntament amb el temps mitjà per època:

- **GCN:** Exhibeix una notable millora al MSE de 2.8701 al Grup 1 a 0.9455 i 0.7926 als Grups 2 i 3, respectivament, indicant una alta efectivitat en l'adaptació a les dades amb el segon temps d'entrenament més ràpid de 10 segons per època.

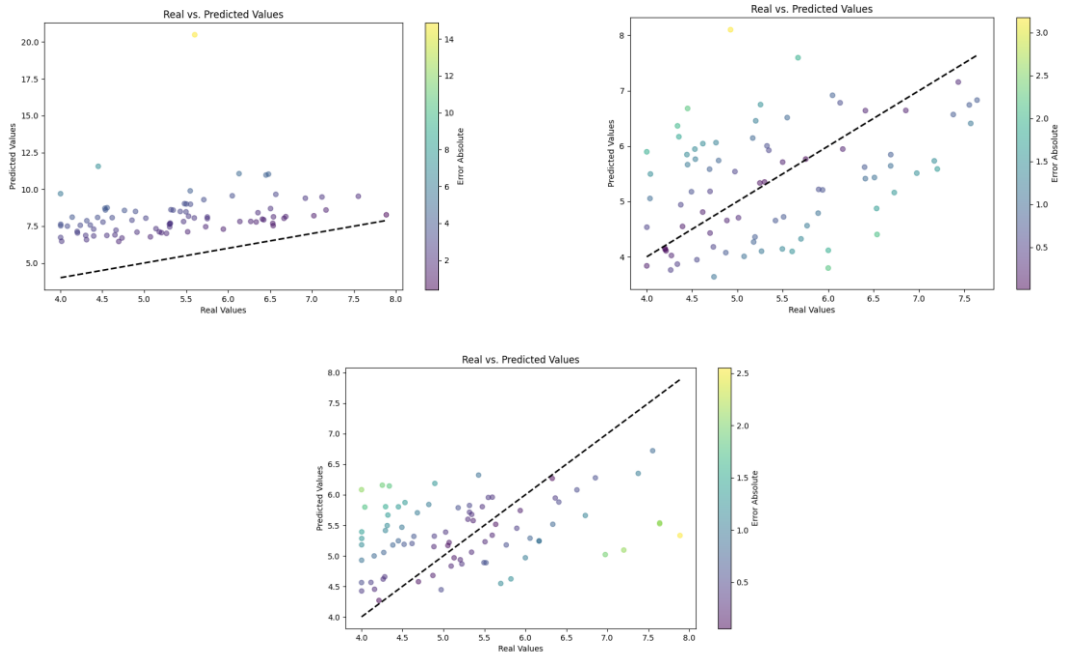


Figura 13. Gràfica que compara els valors predits pel model GCN amb el reals. D'esquerra dreta, grup de dades 1, 2 i 3

- **GT:** Mostra una notable consistència amb MSE relativament baixa i estables a través dels grups (0.7840, 0.8277, 0.834), encara que requereix més temps per època (26 segons), cosa que suggereix que és robust però computacionalment més intensiu.

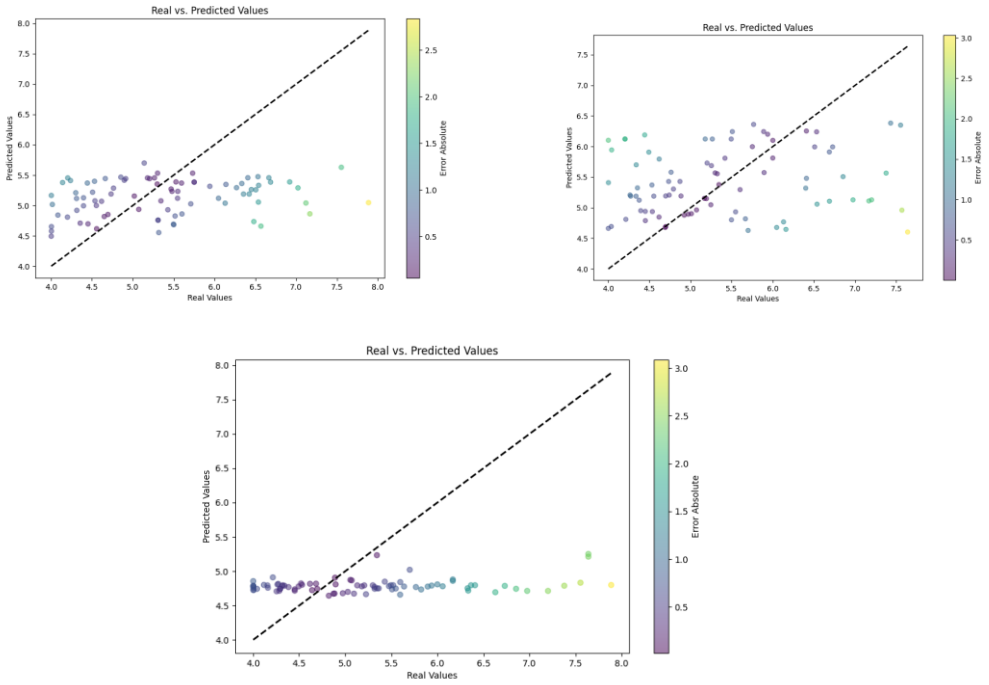


Figura 14. Gràfica que compara els valors predits pel model GT amb el reals. D'esquerra dreta, grup de dades 1, 2 i 3

- **GIN**: Presenta poc canvi als MSEs (2.6637, 2.7103, 2.6212) en els tres grups i el temps per època més curt (3 segons), cosa que indica eficiència en el processament sense grans fluctuacions en el rendiment.

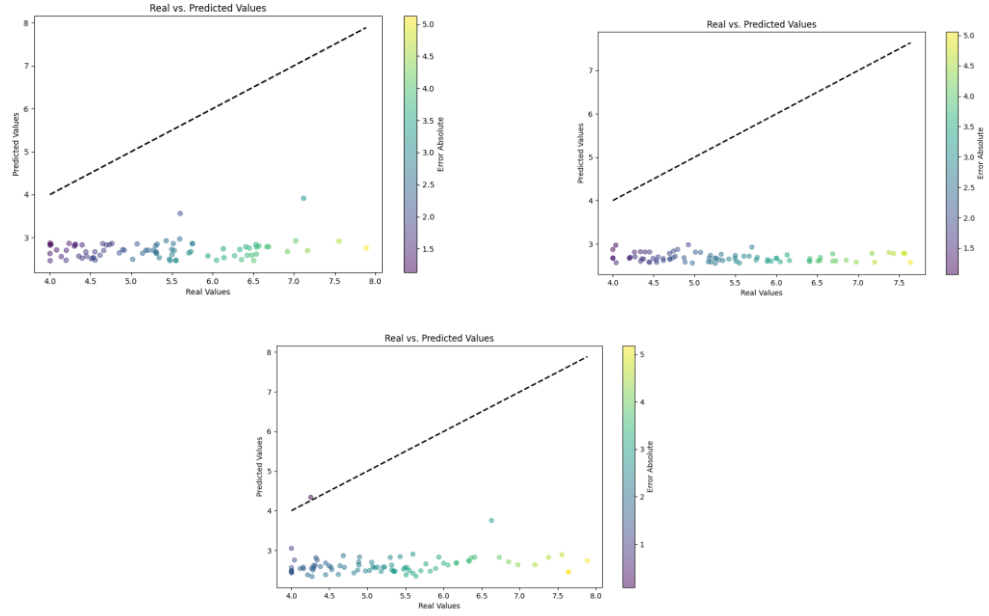


Figura 15. Gràfica que compara els valors predits pel model GIN amb el reals. D'esquerra dreta, grup de dades 1, 2 i 3

- **GatedGCN:** Té una variabilitat més significativa als MSEs (3.1371, 2.8123, 3.6751), amb temps moderats per època (17 segons). Aquest model pot ser sensible a la manera com les dades estan organitzades o segmentades.

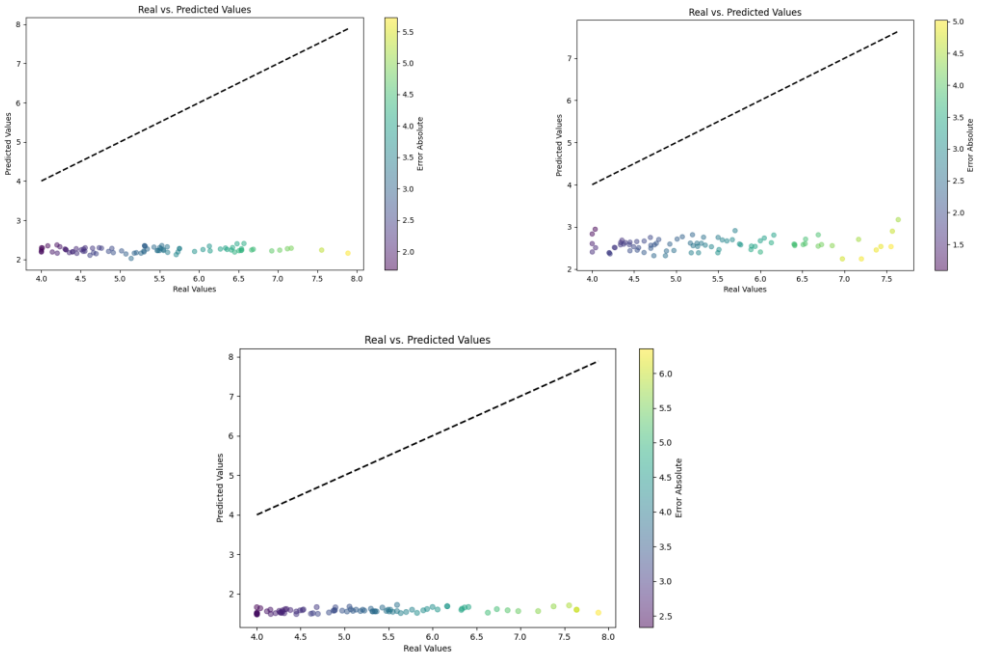


Figura 16. Gràfica que compara els valors predits pel model GatedGCN amb el reals. D'esquerra dreta, grup de dades 1, 2 i 3

- **GAT:** Mostra un increment en MSE de 3.7277 a 3.6992, passant per 3.5489, amb un temps per època de 20 segons. Tot i els seus temps més llargs, el seu rendiment és estable, encara que no mostra millores significatives entre els grups.

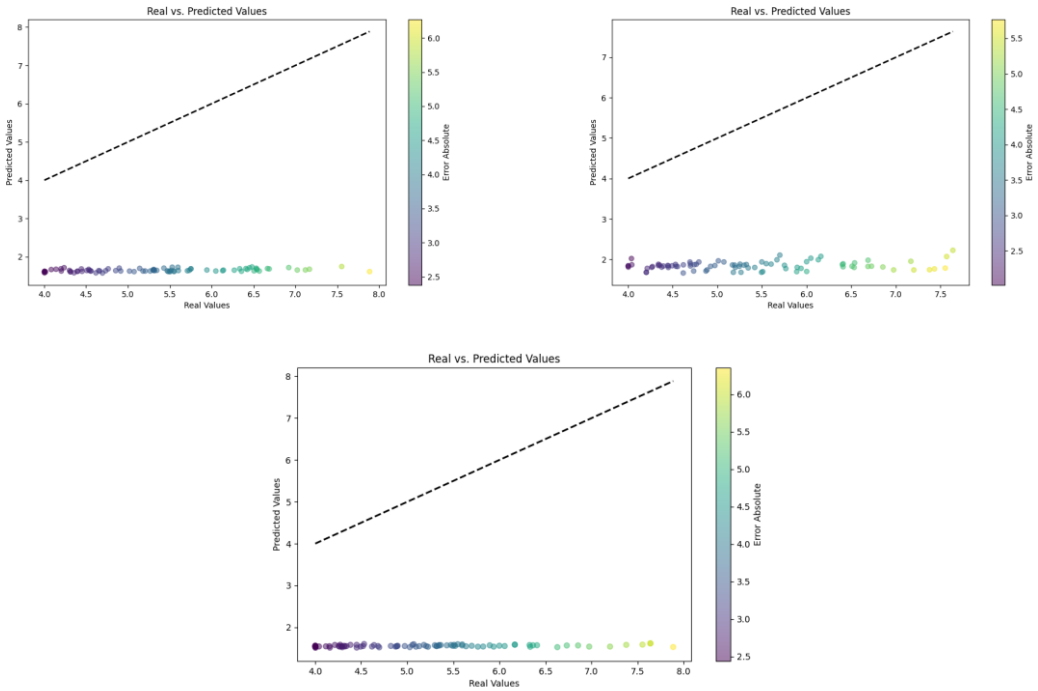


Figura 17. Gràfica que compara els valors predits pel model Gat amb el reals. D'esquerra dreta, grup de dades 1, 2 i 3

En l'avaluació comparativa dels diferents models de GNN, s'observa diferents dinàmiques en termes d'eficiència temporal versus rendiment. El GCN destaca per la seva rapidesa, encara que el GT sobresurt per oferir l'estabilitat més gran en termes de MSE, a pesar de la seva major càrrega computacional. Per altra banda, el GIN ressalta per la seva eficiència operativa, mantenint un rendiment constant amb el menor temps requerit per època.

Addicionalment, tant GT com GIN mostren una robustesa notable, evidenciant consistència en el rendiment a través de diferents conjunts de dades. Aquesta qualitat és especialment crucial en camps on la reproductibilitat és un aspecte fonamental. En contrast, el GatedGCN i el GAT mostren una variabilitat en els resultats que podria indicar una major sensibilitat a quines dades utilitzen. Aquest factor suggereix que es podria necessitar ajustos addicionals en la configuració del model o en la selecció de les dades per optimitzar el seu rendiment.

9 Conclusió

La realització del meu Treball de Fi de Grau (TFG) en Enginyeria Informàtica ha marcat un moment definitiu en la meva trajectòria acadèmica, proporcionant-me una oportunitat única per sintetitzar i aplicar els coneixements i les habilitats desenvolupades durant els anys d'estudi. Aquest projecte no sols ha estat una prova de les meves capacitats tècniques sinó també una plataforma per aprofundir en aplicacions pràctiques d'alta complexitat, especialment en l'àmbit de l'aprenentatge automàtic i la modelització de dades.

Durant el desenvolupament d'aquest TFG, vaig tenir l'oportunitat de treballar amb la biblioteca MolGraph, una eina potent que facilita la implementació de models computacionals complexos. Aquesta experiència no només va enriquir la meva comprensió dels principis fonamentals de les ciències de la computació aplicades a la química computacional, sinó que també em va permetre veure de primera mà com les solucions innovadores poden transformar la recerca científica, permetent anàlisis més ràpides i precises que les metodologies tradicionals.

La complexitat dels desafiaments afrontats en el projecte va exigir un alt grau de pensament crític i adaptabilitat. Vaig aprendre a navegar per problemes de modelatge i anàlisi de dades que requerien no solament una aplicació rigorosa del coneixement teòric sinó també una capacitat per pensar fora dels esquemes tradicionals. La resolució d'aquests problemes va resultar en una millora significativa de les meves habilitats analítiques i tècniques, crucials per al meu desenvolupament professional com a enginyer de software.

L'aspecte de gestió de projectes del TFG també va ser immensament formatiu. Planificar eficaçment les fases del projecte, des de la investigació inicial fins a la implementació final i la validació dels models, va requerir una disciplina i organització estrictes. Aquesta experiència ha reforçat la meva capacitat per gestionar el temps i els recursos, una habilitat inestimable en qualsevol entorn professional on els terminis són crítics i els recursos sovint limitats.

Finalment, el procés d'aquest TFG ha reforçat la meva confiança en la meva capacitat professional i ha solidificat el meu compromís amb una carrera en enginyeria informàtica. A través d'aquest projecte, he confirmat la meva passió per la tecnologia i la innovació i m'he equipat amb una base sòlida per afrontar els desafiaments futurs en la indústria tecnològica.

9.1 Suggeriments per a investigacions futures i millores del sistema

Aquest projecte ha establert una base sòlida per al desenvolupament i validació de models predictius utilitzant la base de dades de la URV amb la biblioteca MolGraph. No obstant això, hi ha diverses àrees en les quals el treball actual pot ser expandit i millorat per a futurs projectes de recerca i desenvolupament.

1. **Expansió de la Capacitat de l'Entrenament de Models:** El codi desenvolupat per a l'entrenament dels models ofereix una estructura flexible que pot ser ampliada per incloure un ventall més ampli de paràmetres configurables per a cada tipus de capa. Aquesta expansió podria incloure paràmetres com el tipus de normalització (per exemple, batch normalization, layer normalization), diverses funcions d'activació, o

diferents mètodes de regularització. Aquestes modificacions permetrien als investigadors adaptar més finament els models a les necessitats específiques dels seus conjunts de dades i objectius d'investigació, millorant la precisió i l'eficàcia dels models en predir propietats moleculars.

2. **Incorporació de Noves Arquitectures de Capes:** El fitxer “layers”, que actualment conté les definicions de les capes utilitzades en els models, està dissenyat per ser extensible. Es recomana que investigadors futurs aprofitin aquesta estructura per experimentar amb noves arquitectures de xarxes neuronals que poden emergir a mesura que avança la investigació en el camp dels grafs i l'aprenentatge profund. Això pot incloure l'exploració de noves variants de capes de Graph Neural Networks (GNNs) que podrien oferir millors rendiments o que estan especialment dissenyades per a tipus específics de dades moleculars.
3. **Modificació del Tipus de Decodificació:** El fitxer “encoder” proporciona una base per a futures investigacions que desitgin modificar o millorar el tipus de decodificació utilitzat per transformar les dades en formats que siguin més adients per al processament per models de deep learning. Investigacions futures podrien explorar diferents estratègies de codificació que poden millorar la manera en què les propietats moleculars són interpretades i utilitzades pel model, potencialment millorant la precisió de les prediccions.
4. **Generalització per a Altres Bases de Dades:** Mentre que els codis actuals estan optimitzats per a la base de dades de la URV, s'encoratja a futurs projectes a adaptar i provar aquests models en altres conjunts de dades moleculars. Això no només augmentaria la utilització pràctica dels codis sinó que també ajudaria a validar la generalitzabilitat i robustesa dels models en una àmplia varietat de contextos científics i aplicacions reals.

Aquestes àrees d'expansió no solament proporcionen oportunitats per avançar en el camp de la química computacional i la bioinformàtica sinó que també potencialitzen la col·laboració interdisciplinària i el desenvolupament de noves tecnologies que poden tenir un impacte significatiu en la societat i la indústria. Aquest projecte estableix un marc que, amb la inversió contínua en investigació i desenvolupament, pot conduir a descobriments transformadors en el camp de l'enginyeria informàtica aplicada a les ciències de la vida.

10 Bibliografía

Alexander Kensert, G. D. (4 de setembre de 2023). *MolGraph: a Python package for the implementation of molecular.* Obtenido de https://arxiv.org/pdf/2208.09944

Grinberg, D. (2 de Agost de 2023). *An introduction to graph theory.* Obtenido de https://arxiv.org/pdf/2308.04512

Haohan Wang, B. R. (2017, Març 3). *On the Origin of Deep Learning.* Retrieved from arXiv: https://arxiv.org/pdf/1702.07800

Justin Gilmer, S. S. (12 de juny de 2017). *Neural Message Passing for Quantum Chemistry.* Obtenido de https://arxiv.org/pdf/1704.01212

Kipf, T. N. (22 de febrer de 2017). *SEMI-SUPERVISED CLASSIFICATION WITH.* Obtenido de https://arxiv.org/pdf/1609.02907

Mcinerney, A. (14 de Novembre de 2023). *Feedforward neural networks as statistical models.* Obtenido de https://arxiv.org/pdf/2311.08139

SDF file. (s.f.). Obtenido de https://www.nonlinear.com/progenesis/sdf-studio/v0.9/faq/sdf-file-format-guidance.aspx

Tello, D. J. (13 de Juny de 2018). *arXiv.* Obtenido de https://arxiv.org/pdf/1806.05298

Vaswani. (2 de Agost de 2023). *Attention Is All You Need.* Obtenido de https://arxiv.org/pdf/1706.03762

Velickovi, P. (4 de Febrer de 2018). *GRAPH ATTENTION NETWORKS.* Obtenido de https://arxiv.org/pdf/1710.10903

Vijay Prakash Dwivedi, X. B. (24 de Gener de 2021). *A Generalization of Transformer Networks to Graphs.* Obtenido de https://arxiv.org/pdf/2012.09699v2

Xu, K. (22 de febrer de 2019). *HOW POWERFUL ARE GRAPH NEURAL NETWORKS?* Obtenido de https://arxiv.org/pdf/1810.00826v3

11 Annexes

En aquest apartat s'explica detalladament el procés per posar en marxa el programa desenvolupat durant el treball de fi de grau. A continuació, es descriuen els passos necessaris per a la instal·lació, l'ús i els requisits del sistema.

Instal·lació de la Llibreria MolGraph: Per començar, és imprescindible tenir instal·lada la llibreria MolGraph, que és el nucli del nostre sistema de modelatge. MolGraph es pot instal·lar de diverses maneres; en aquest projecte, s'ha optat per la instal·lació mitjançant Docker, seguint les instruccions proporcionades en la documentació oficial de MolGraph disponible en el seu repositori de GitHub. Aquest mètode garanteix que l'entorn de treball conté totes les dependències necessàries per al correcte funcionament de la llibreria sense conflictes amb altres paquets.

Configuració del Contenidor Docker: Un cop instal·lada la llibreria MolGraph dins d'un contenidor Docker, el següent pas consisteix a preparar l'entorn per als scripts del projecte. Això s'aconsegueix copiant els fitxers necessaris del projecte dins del contenidor Docker utilitzant el comandament:

```
Docker cp <path_fitxer_local> <nom_del_contenidor>:/path_en_contenidor
```

Aquest procés assegura que tots els scripts i dades necessàries estan disponibles dins de l'entorn Docker per ser executats.

Execució del Programa: Per executar el programa, primerament s'ha de preparar un script principal, que actuarà com a punt d'entrada per a l'execució dels diferents mòduls del projecte. Aquest script invocarà els diferents fitxers de Python que constitueixen el projecte, gestionant el flux de dades i la seqüència d'operacions de modelatge.

Finalment, per començar l'execució del programa, només cal obrir una terminal dins del contenidor Docker i executar el script amb la següent comanda:

```
$ python3 script.py
```

Aquest comandament inicia el procés de modelatge, des de la càrrega i preparació de dades fins a l'entrenament i validació dels models, segons s'especifiqui en el script.

Requeriments del Sistema: El sistema on s'executa el contenidor Docker ha de complir amb les especificacions mínimes requerides per a executar aplicacions de deep learning, inclòs una quantitat adequada de memòria RAM i, idealment, accés a una GPU per accelerar els càlculs de l'entrenament de models. També és essencial disposar d'una bona connexió a internet durant la instal·lació per descarregar les imatges de Docker i les dependències necessàries.

Aquest apartat dels annexos serveix com a guia detallada per a usuaris i desenvolupadors que desitgin replicar o expandir el treball fet en aquest projecte de fi de grau, proporcionant una base sòlida per a la continuació de la recerca i el desenvolupament en l'àmbit de la modelització computacional amb MolGraph.