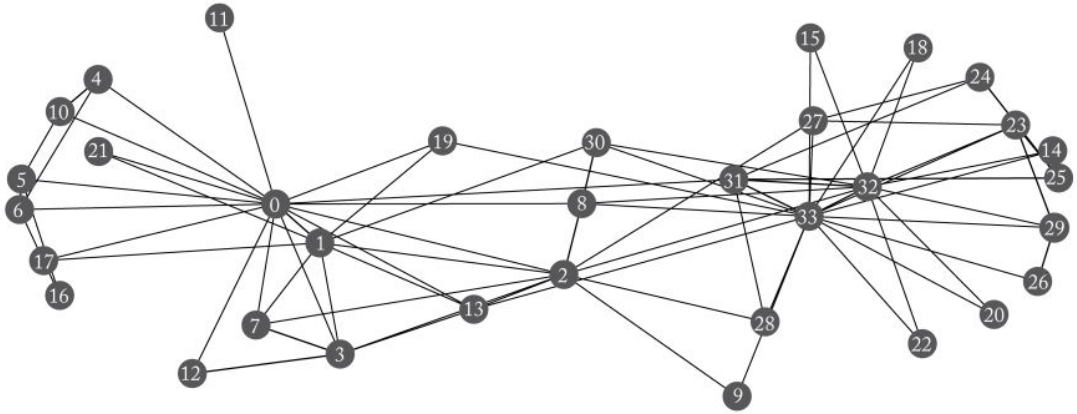
GNNs to predict physical properties

Otaviano Cruz - Mestrado/CAP

Outline

- Graphs
- GNNs
- QM9
- Methodology
- Results
- Conclusions

// Graphs

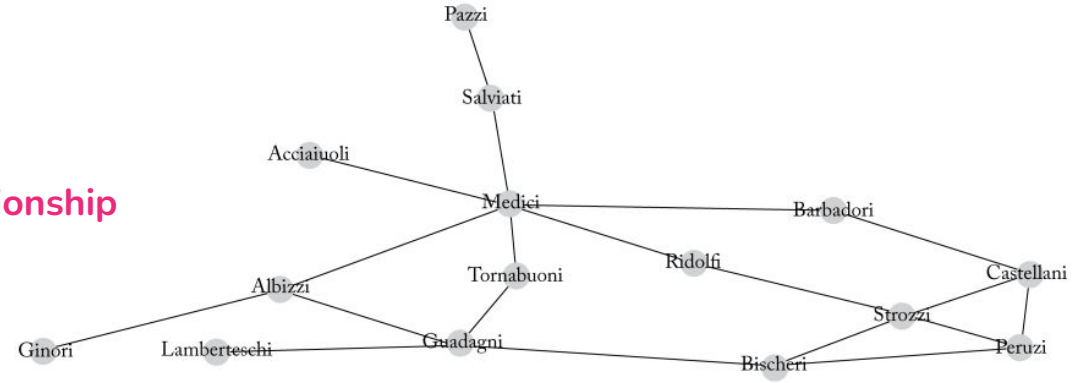


Points

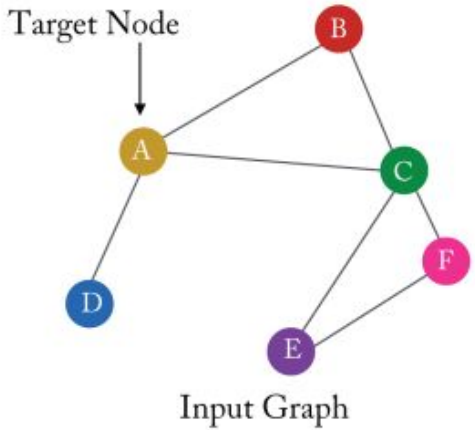
Representation Problems

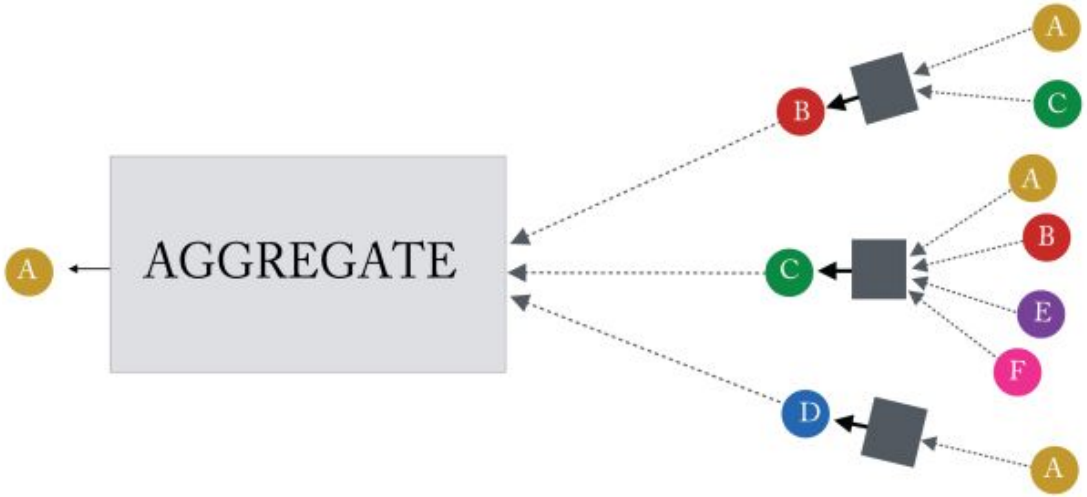
- Adjacency Matrix
- Invariances

Family relationship



// GNNs

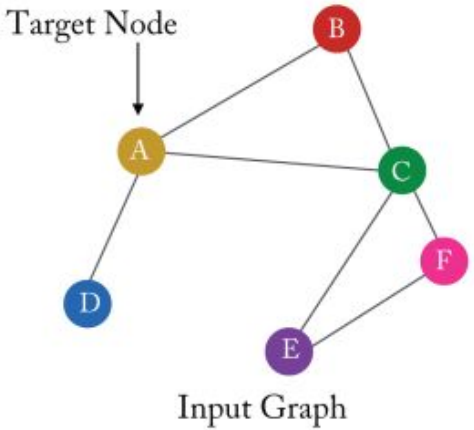


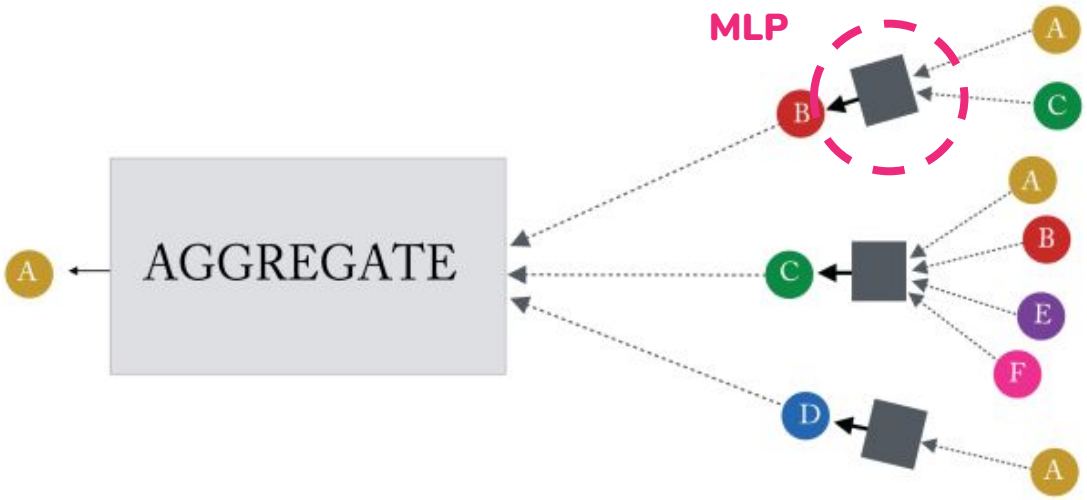


Node A receive a message from the neighborhood around a influence area.

Neural Message Passing
[1]

// GNNs





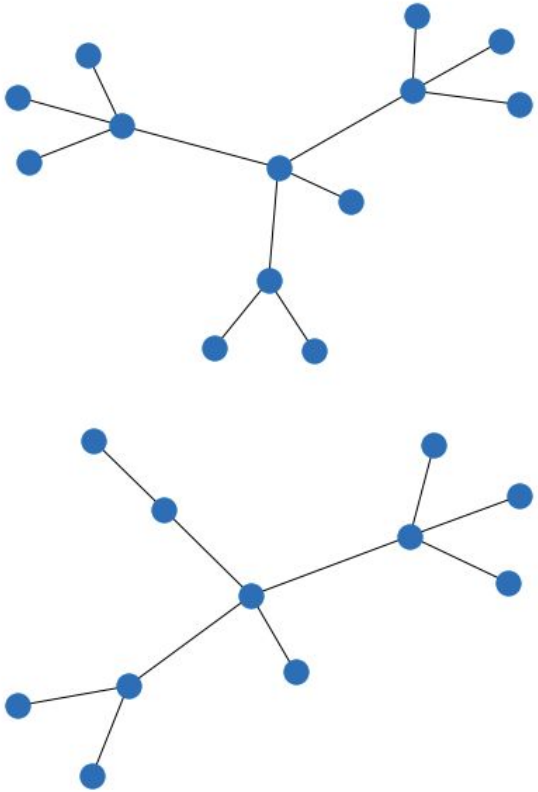
But, what we want to learn using this model?

- **Structural Information (benzene rings)**
- **Features Properties (atom types)**

Neural Message Passing

[1]

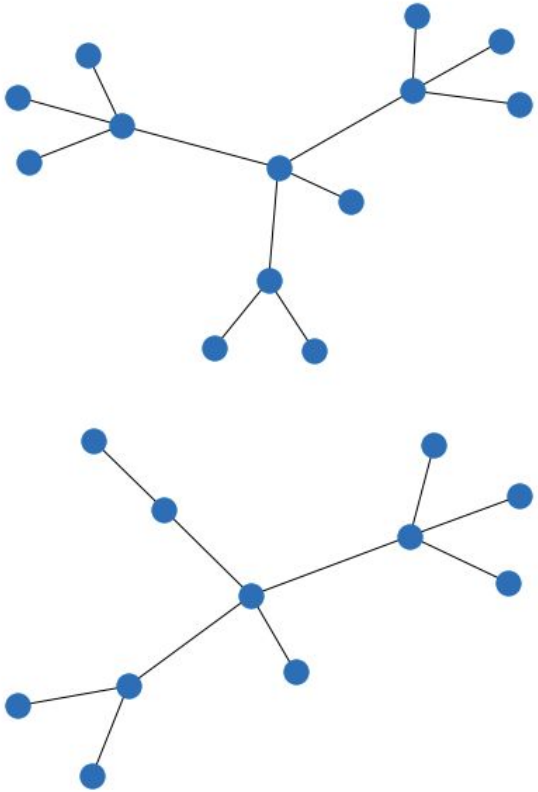
// QM9

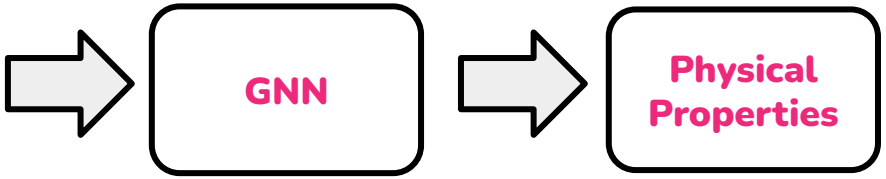


With approx. 134K molecules, the QM9 Dataset [2] provide 19 physical properties using Molecular Dynamics *ab initio* calculations:

- **Dipole Moment.**
- **Isotropic polarizability.**
- **Band Gap.**
- **Internal Energy.**
- **...**

// Methodology





Supervised Paradigm :

- **Loss Function: RMSE**
- **Activation Function : ReLu**
- **Train/Test: 70%/30%**

// Methodology

Grid Search (Hyper Parameters Optimization)

Tools:

- Optuna (Instances Parallelism).
- Tensorboard (Compile all the instances).
- Torch Geometric (Modeling).

Hyperparameters Evaluation:

- Convolutional Layers: 1 - 4
- Hidden FC Layers: 1 - 3
- Hidden Nodes: 2 - 128
- Batch Size: 1 - 1024
- Learning Rate: 1e-5 - 1e-1

How to select the best model ?

- Execution Time.
- Validation Error.
- Train Error.

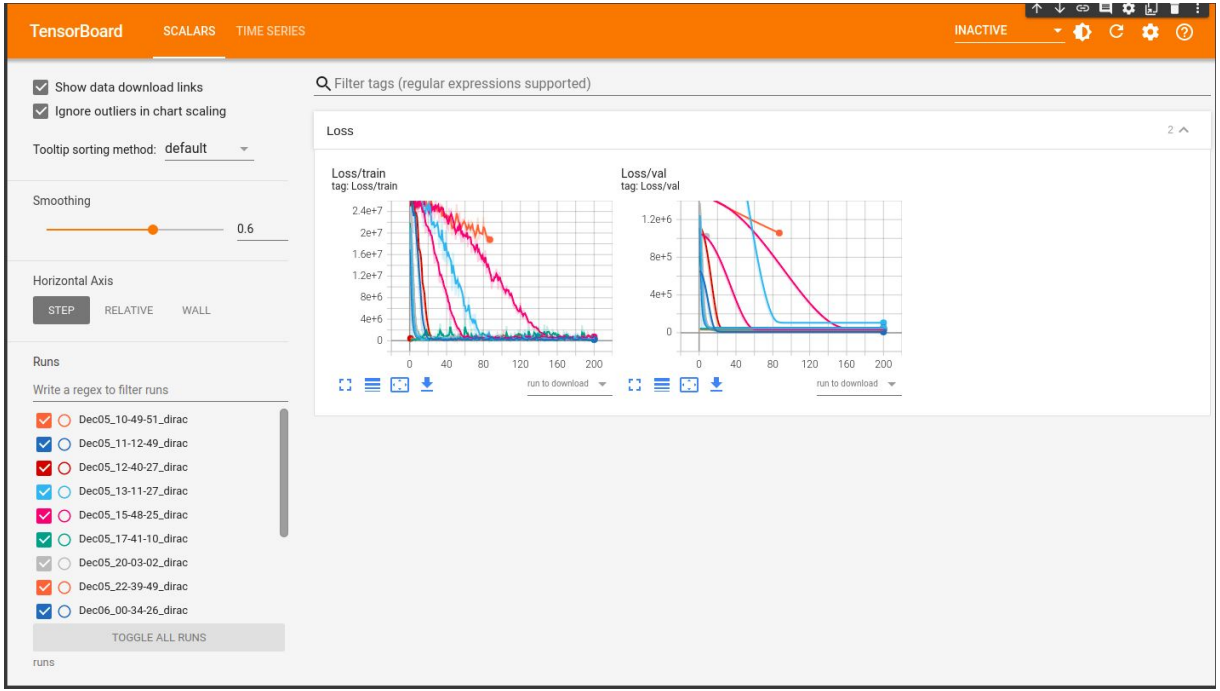
Hardware:

CPU: Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz
RAM: 755GB
GPU: 4xNVIDIA Tesla V100 16GB

// Results

Tensorboard Dashboard:

- **Train loss.**
- **Validation loss.**



Obs: Gradient Vanish/Explosion problem

// Results

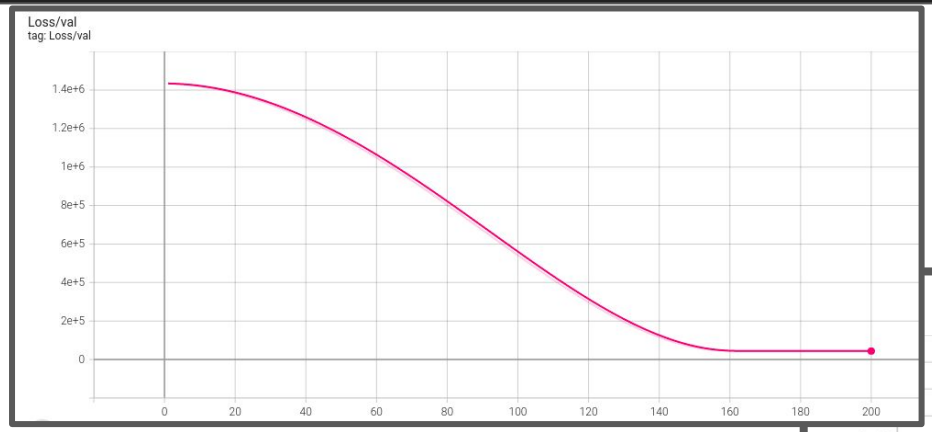
Best Model

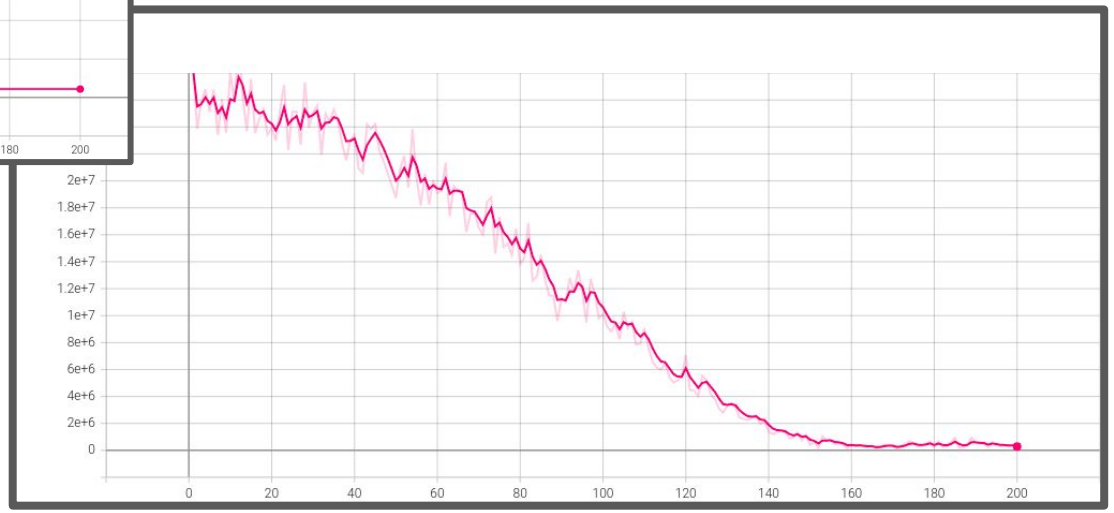
- Convolutional Layers: 4
- Hidden FC Layers: 4
- Hidden Nodes: 64
- Batch Size: 128
- Learning Rate: ~0.001
- Trainable params: 186707

Metrics

- Train Loss: 3724.4
- Validation Loss: 2538
- Execution Time: 1h 53min

// Results


Loss/val
tag: Loss/val
1.4e+6
1.2e+6
1e+6
8e+5
6e+5
4e+5
2e+5
0
0
20
40
60
80
100
120
140
160
180
200


2e+7
1.8e+7
1.6e+7
1.4e+7
1.2e+7
1e+7
8e+6
6e+6
4e+6
2e+6
0
0
20
40
60
80
100
120
140
160
180
200

// Conclusions

Attention Points

- **Unsatisfactory results.**
- **1.800 model rateds.**
- **Recurrent Gradient Vanishing.**
- **Under and Over Fitting problems.**
- **Optimization problems.**

Next steps

- **Evaluate others architecture models.**
- **Include the tensorboard gradient diagrams to create a early stopping based on.**
- **Try to use cuGraph, cuDNN and DGL to increase the performance and optimize GPU calculations.**
- **Try to translate classical calculations in MD *ab initio* to GNN (MLFF).**

Thank You

otavianocruz@id.uff.br

// References

[1] Neural Message Passing - http://arxiv.org/abs/1704.01212

[2] QM9 Dataset - https://arxiv.org/abs/1703.00564

Documentation: Pytorch / Torch Geometric / Optuna / Tensorboard