INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

INP TOULOUSE ENSEEIHT

Prédiction des énergies de molécules
S10 ModIA

Ababacar CAMARA, Sébastien GRAND

Juin 2022

Table des matières

Introduction

Nous présentons dans ce rapport l'ensemble des tâches réalisées en vue de prédire l'énergie potentielle d'une molécule compte tenu de ses composants atomiques et de leur disposition géométriques (dénotés $r = r_1, r_2, , r_N$) en appliquant du machine learning. Dans ce cadre, notre modèle devra remplir la condition d'invariance par symétrie. Plus précisément, la prédiction de l'énergie d'une molécule donnée devra donner le même résultat quelque soit la translation, la rotation ou la permutation subie par cette dernière. Pour répondre à cette problématique, nous allons d'abord faire de l'exploration de donnée et faire le choix d'une représentation de données (pré-traitement des données), ensuite nous présenterons la méthodologie utilisée pour résoudre le problème. Enfin, nous interpréterons les résultats obtenus.

1 Exploration et visualisation des données

Pour le chargement des données, nous avons utilisées la librairie ase, cette dernière nous permet également de récupérer d'autre informations utiles à la résolution du problème, comme la valence des atomes, la masse des molécules et le numéro atomique des molécules et entre autre les visualiser voir ci dessous par exemple :

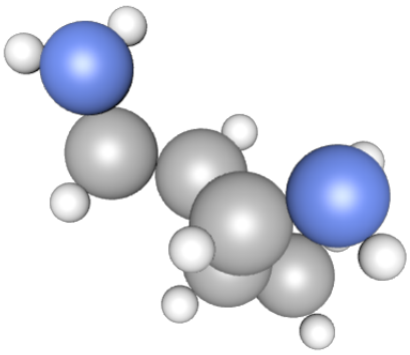

FIGURE 2 – Visualisation 3D par ASE de la molécule C5H12N2

Après avoir chargées les données, nous avons réalisé quelque basique statistiques, dont les principales observations sont indiquées ci-dessous :

— Le nombre d'individus dans le jeu d'entraînement dont on a le fichier .xyz et l'énergie associée est du nombre de 6770. Le nombre d'individus dans le jeu de test 1689.
— Certaines molécules présent dans le jeu de test ne sont pas présents dans le jeu d'entraînement. Cependant, les atomes C,H,N,Cl,O,S sont aussi bien présents dans le jeu d'entraînement que de tests.
— Le nombre de molécules différentes dans le jeu d'entraînement est de 107 et celle dans le jeu de donnée de test est de 78
— L'énergie est linéairement corrélée au nombre d'atomes présentes dans la molécule.

— L'énergie dépend de la masse de la molécule. voir annexe

2 Pré-traitement des données : Choix de la représentation des données et transformation réalisées.

2.1 Matrice de Coulomb et matrice Alpha

Dans cette partie, nous nous sommes appuyés sur [1], qui présente une approche constituée de deux types de données d'entrées. La première donnée d'entrée est la matrice de Coulomb calculée à l'aide de la librairie `dscribe`, elle est un simple descripteur global qui imite l'interaction électrostatique entre les atomes. Elle est définie comme suit :

$$M_{ij}^{Coulomb} = \begin{cases} 0.5Z_i^{2.4} \text{ for } i = j \\ \frac{Z_i Z_j}{R_{ij}} \text{ for } i \neq j \end{cases}$$

Les éléments diagonaux peuvent être considérés comme l'interaction d'un atome avec lui-même et sont essentiellement un ajustement polynomial des énergies atomiques à la charge nucléaire Z_i. Les éléments hors diagonale représentent la répulsion de Coulomb entre les noyaux $i$et j[2]. Ensuite, nous calculons le spectre des valeurs propres de la matrice de coulomb en résolvant le problème $Cv = \lambda v$ sous les contraintes que $\lambda_i >= \lambda_{i+1}$ et ou $\lambda_i > 0$. On obtient ainsi un signal 1D composé de valeurs propres pour chaque molécule. Pour finir, on applique un Z-score sur les données afin que chaque colonne soit un vecteur de moyenne nulle et de varaince 1.

La seconde données d'entrée est une matrice α dont chaque colonne se réfère à un atome, ici du jeu de données $(C, H, N, O, S$ ou $CL)$ et chaque ligne se réfère à une molécule du jeu de données (ex. CH_4). Ainsi, le vecteur associé à la molécule CH_4 sera $[1\,4\,0\,0\,0\,0]$.

On appliquera en entrée d'un réseau bayésien une concaténation des deux matrices.

2.2 Many Body Tensor Representation.

Pour la représentation des molécules, nous allons prendre en compte le nombre d'atomes , la masse, le nombre d'éléments de chaque atomes dans la molécule (C,H,N ...) et enfin la une représentation de [3] nommée Many Body Tensor Representation (en utilisant la librairie dscribe). Cette représentation globale de la molécule a pour avantage de respecter l'invariance par symétrie et donne de meilleurs résultats notamment en comparaison avec la matrice de coulomb. Cette matrice se base sur la matrice de couloumb CM qui est une matrice de distance atome par atome pondérée avec leur numéro atomique.($M_{i,j} = Z_i Z_j / d_{i,j}$ en dehors des diagonales et $0.5Z_i^{2.4}$ au niveau des diagonales. et avec Z_i le numéro atomique et $d_{i,j}$ la distance euclidienne entre deux atomes) En se basant sur cette matrice, l'invariance par translation et par rotation est garantie. Pour s'affranchir de l'ordre des atomes et donc assurer une invariance à la permutation. La méthodologie adopté se base sur le BoB [7]. Ainsi pour

chaque élèment on obtient ainsi la formule suivante :

$$f_k(x,z) = \sum_{i=1}^{N_a} w_k(i) D(x, g_k(i)) \prod_{j=1}^{k} C_{z_j, Z_{i,j}}$$

où $z \in N^k$ sont des nombres atomiques, $i = (i_1, ..., i_k) \in 1, ..., {N_a}^k$ sont des tuples d'index, et w_k , g_k assignent un scalaire à k atomes dans la matrice de coulomb CM

2.2.1 Paramètrage de la fonction MBTR de Dscribe

Nous avons fixé les hyperparamètres k1,k2,k3 de la façon suivante : k1 a pour fonction le numéro atomique, k2 l'inverse de la distance et k3 la fonction cosinus. La taille de grille n=5 avec un bruit sigma =0.1 et pour seuil $1e-3$. Les autres paramètres ont été prise par défaut, en considération de l'exemple sur le site. Sous cette configuration la sortie de la représentation vectorisé MBTR à pour taille (765,1). La visualisation d'une molécule par rapport à cette représentation peut être vue sur la figure suivante :

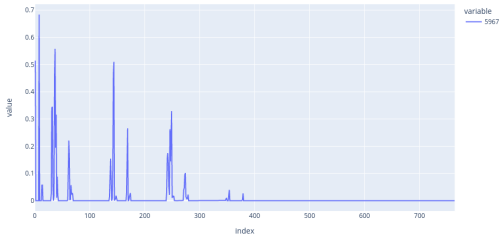

FIGURE 3 – Énergie par rapport à la masse de la molécule

2.3 Scattering representation

Dans cette partie, nous avons souhaité changer la représentation des données. Pour cela, nous nous sommes appuyés sur [4][5][6] afin d'utiliser un `Scattering Transform`, qui est une transformation linéaire qui nous permet de construire une invariance aux transformations géométriques. Ce que nous souhaitons faire car nous voulons ne pas être sensible aux rotations ou translations. Ainsi, à l'aide de la transformation 3D, on devient invariants aux translations et rotations. Le `Scattering Transform` est défini comme un réseau neuronal convolutif à valeurs complexes dont les filtres sont des ondelettes et la non linéarité est un module complexe. Chaque couche du réseau est une transformée en ondelette. La séparation des échelles par les ondelettes permet également une stabilité à la déformation de la données initiales. Ainsi, cette transformation semble bien adaptée à notre problème car les molécules peuvent être vues comme des signaux structurés.

2.3.1 Données d'entrée pour le `Scattering Transform`

Tout d'abord avant le calcul de transformée, nous créons trois types de données, les charges nucléaires complètes, le nombre d'électrons de valence que l'on stocke comme la charge de valence et les positions des atomes de la molécules. Ensuite on normalise les positions des atomes, puis on défini, la transformée en utilisant HarmonicScattering3D, avec pour paramètre $J = 2$. Puis par batch de 16, on crée `order_0` et `order_1_2` résultats de la fonction *compute_integrals*, qui correspondent aux coefficients d'ordre zéro et d'ordre un et deux. De là on obtient, les coefficients de la transformée des données de charges, de valence et de la différence entre les deux `core`. Pour chaque batch on stocke les coefficients dans un tenseur.

2.4 Configuration pour l'entraînement

Après pré-traitement, la taille du jeu de donnée est de (6770, 773). Nous avons divisé ce jeu de donnée en jeu d'entrainement(95%) et de test(5%) en utilisant la fonction train test split de sklearn et avec la graine 42. Ensuite pour leur application sur des algorithmes d'apprentissage superficiel, elles ont été normalisée. Cependant, elles ont été laissé tel quel pour les réseaux de neurones.

3 Application sur modèles de Machine Learning.

Les meilleurs résultats ont été obtenus en utilisant des réseaux de neurones. Cependant, avant de passer à ces derniers nous avions d'abord testé quelques algorithmes de machine learning de base avec la représentation MBTR. Les résultats sont consignés sur le tableau suivant :

Algorithmes utilisés			
Noms	RMSE Train	RMSE Test	Hyperparamètres
XGboost	0.0069	0.5653	reg_alpha=0.8
RandomForestRegressor	0.12	0.77	default
SVM	0.16	0.34	kernel = "linear"
Ridge Regression	0.15	0.29	alpha=0.8

On en déduit que certains des paramètres de notre problème ont des intéractions linéaires.

4 Application du modèle sur modèle de Deep Learning.

Afin d'obtenir les structures de nos réseaux de neurones nous avons utilisé la méthodologie suivante :

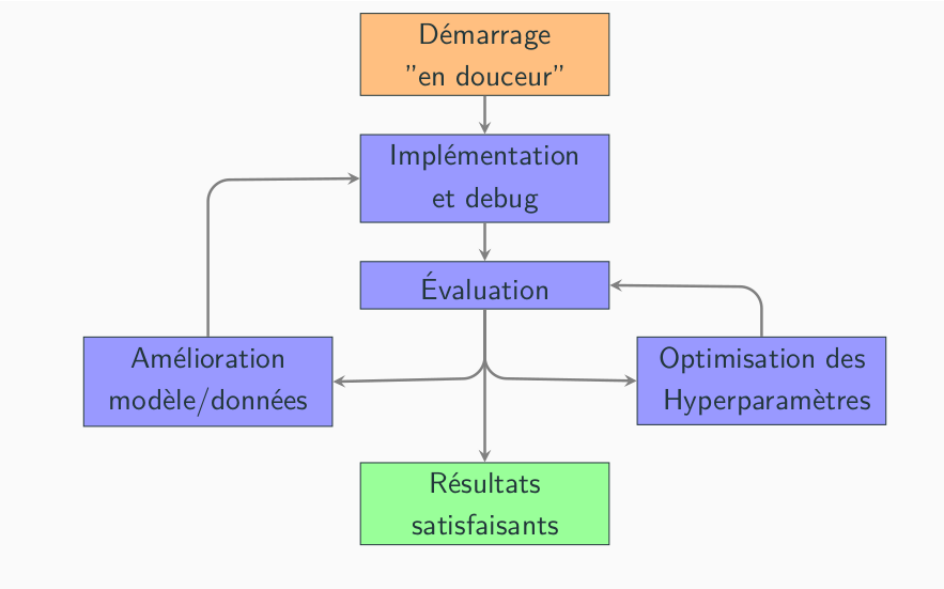

FIGURE 4 – Méthodologie de Josh Tobin. Image tirée du cours Apprentissage Statistique en Grande dimension d'Axel Carlier

Ainsi, nous avons au cours de nos recherches pu optimiser nos résultats pour tenter d'atteindre de bons scores avec nos différentes représentations.

4.1 Bayesian Neural Network : Application à la matrice alpha et la matrice de Coulomb

Le réseau utilisé pour l'utilisation de la matrice de Coulomb couplée à la matrice α est un réseau bayésien, qui prend en entrée un vecteur de taille 276, et qui est composé (pour le meilleur réseau) de trois couches, la première contenant 18 neurones, la seconde 9 neurones et la dernière 3 neurones. Le schéma simplifié est présenté ci-dessous.

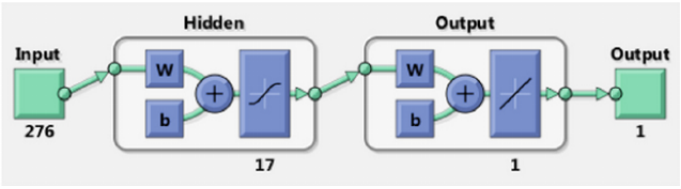

FIGURE 5 – Modèle simplifié du Bayesian Neural Network utilisé (image issu de [1])

Les résultats obtenus sont listés dans le tableau ci-dessous :

Algorithmes utilisés		
Modèles (Neurones par couche)	RMSE sur nos données	Score submit
BNN (16)	0.78	/
BNN ($16 \times 8 \times 4$)	0.53	/
BNN ($18 \times 9 \times 3$)	0.19	0.40

Les résultats sont un peu décevants, cependant, agrandir le réseau n'était pas possible car notre mémoire n'était pas assez importante, notre capacité de calcul non plus. Ainsi, nous avons opté pour un changement de représentation des données, nommé *Many Body Tensor Representation*.

4.2 Convolutional Neural Network : Application MBTR

Afin de suivre la méthodologie, nous avons d'abord commencer par le plus simple c'est-à-dire entraîner un réseau de neurones dense "fully connected". La première structure définie comportait juste 32 neurones puis 1 neurone en sortie. Cependant, le modèle sous-apprenait. Ensuite, en augmentant progressivement le nombre de neurones, et la profondeur du jeu de donnée, nous sommes parvenus à atteindre une loss de 0.13. Il faut également noté, que le choix de la fonction d'activation Relu a été primordial à l'amélioration des résultats. Pour partir plus loin, nous avons utilisé des couches convolutives. Et cela nous a permis de descendre jusqu'à une loss de 0.07. La structure finale obtenue est la suivante : elle est constituée de 3 couches de convolution de 1D (64,128,256), suivies de couches linéaires (500 puis 1) qui nous renvoie une prédiction d'énergie d'atomisation. Le schéma simplifié du modèle est présenté c-dessous.

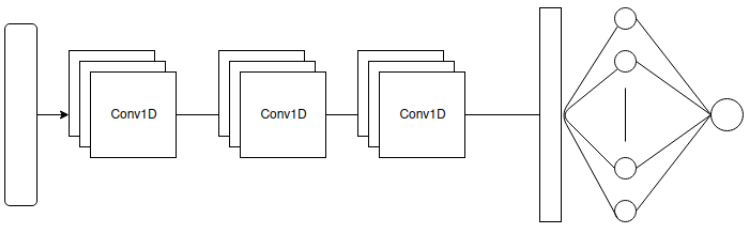

FIGURE 6 – Modèle CNN simplifié utilisé

Les résultats sont présentés dans le tableau ci-dessous :

Algorithmes utilisés		
Noms (paramètres)	RMSE sur nos données	Score submit
FNN	0.13	0.12
CNN	0.07	0.08

4.3 Fully Connected Neural Network & Multilinear regression : Application au Scattering

Le modèle Fully Connected est présenté ci-dessous dans une version simplifiée.

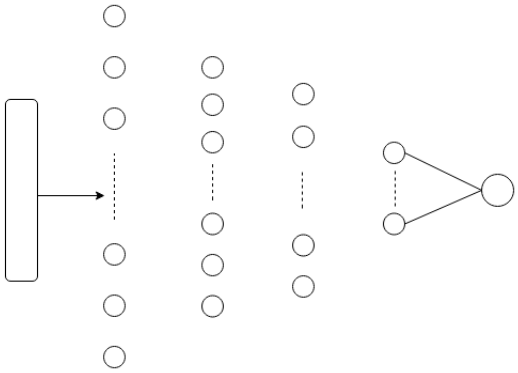

FIGURE 7 – Modèle Fully Connected simplifié utilisé

La formule mathématique de la *Multilinear Regression* est définie ci-dessous :

$$\tilde{E}_r(\rho_x) = b + \sum_i (\nu_i \prod_{j=1}^{r} (\langle S\rho_x, w_i^{(j)} \rangle + c_i^{(j)})).$$

FIGURE 8 – Formule Multilinear Regression [6]

Ici, i correspond au nombre de couche que l'on aura dans le modèle, r correspond au nombre de couche *linear* que l'on a dans i-ème couche. Nous avons choisi de prendre $i = 10$ et $r = 3$ pour notre modèle. La vecteur d'entrée du réseau est définie par $S_{\rho x}$.
Les résultats du pour les données de la transformée pour chaque modèle sont présentés dans le tableaux ci-dessous :

Algorithmes utilisés		
Noms (paramètres)	RMSE sur nos données	Score submit
Multilinear regression (i=5, r=3)	0.17	0.28
Fully connected Network ($128 \times 64 \times 32 \times 8$)	0.15	0.31
Linear Regression	0.35	0.48

On peut donc voir que nos résultats ne sont pas si satisfaisants même avec une bonne transformation de données, il convient peut être de changer des paramètres de calcul du scattering notamment la valeur de J, pour ajuster au mieux les données mais le manque de capacité de calcul pour le scattering rend la tâche difficile. Cependant, l'utilisation de la *Multilinear Regression* permet d'améliorer le score obtenu avec un simple régression linéaire.

Conclusion

Ce concours Kaggle et ce travail sur un jeu de données de molécules dont le but était de prédire l'énergie d'atomisation à pour nous été l'occasion de travailler sur un jeu de données que nous ne connaissions pas et un domaine totalement inconnu. Ainsi, nous avons dû nous adapter aux données et trouver différentes représentations afin de les exploiter au mieux et obtenir de bons résultats. Puis, nous avons expérimentés plusieurs modèles de Machine et Deep Learning. Au final, c'est l'approche *Many Body Tensor Representation* qui remporte les meilleurs résultats.

5 Annexe

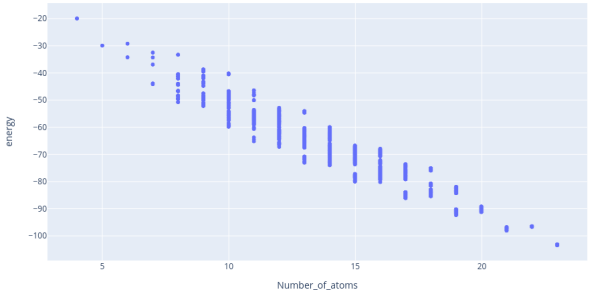

FIGURE 9 – Énergie par rapport au nombre d'atomes

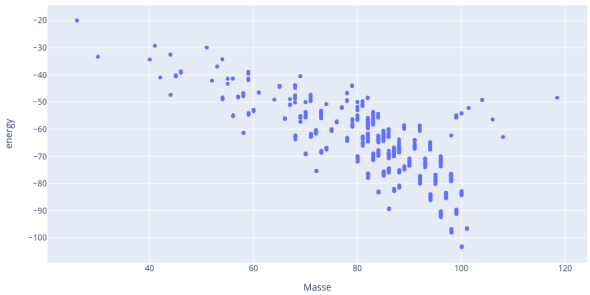

FIGURE 10 – Énergie par rapport à la masse de la molécule

6 Références

[1]. Tchagang.A and Valdés.J 2019 Prediction of the atomization energy of molecules using Coulomb matrix and atomic composition in a Bayesian regularized neural networks ;

[2]. https ://singroup.github.io/dscribe/latest/

[3]. Haoyan Huo and Matthias Rupp. Unified representation of molecules and crystals for machine learning. arXiv e-prints, pages arXiv :1704.06439, Apr 2017. arXiv :1704.06439. [4]. Hirn, M., Poilvert, N. Mallat, S. Quantum energy regression using scattering transforms ;

[5].Hirn, M., Mallat, S. Poilvert, N. Wavelet scattering regression of quantum chemical energies.

[6]. Eickenberg, M., Exarchakis, G., Hirn, M., Mallat, S. Thiry, L. Solid harmonic wavelet scattering for predictions of molecule properties.

[7]. K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K.-R. Müller, A. Tkatchenko, Ma- chine learning predictions of molecular properties : Ac- curate many-body potentials and nonlocality in chemical space. J. Phys. Chem. Lett. 6, 2326–2331 (2015).