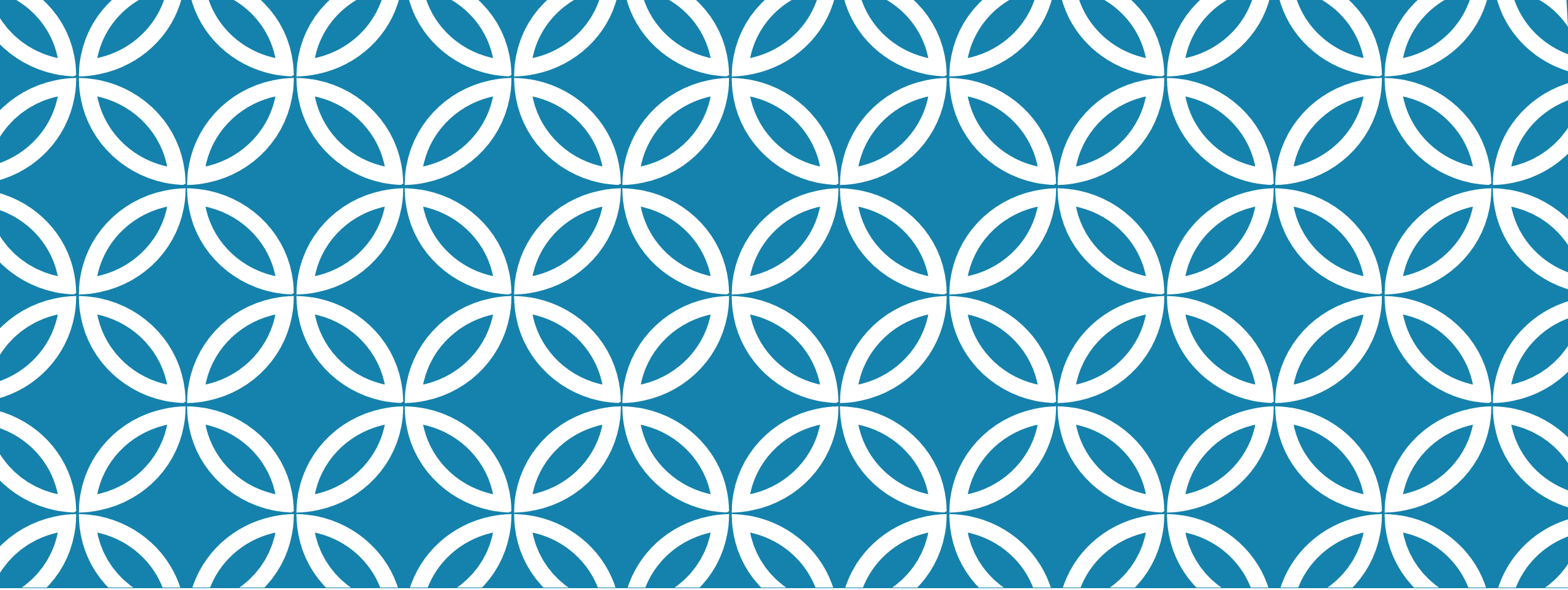


# 機械学習ポテンシャル

実践編

分子科学研究所　南谷英美

# 機械学習ポテンシャル

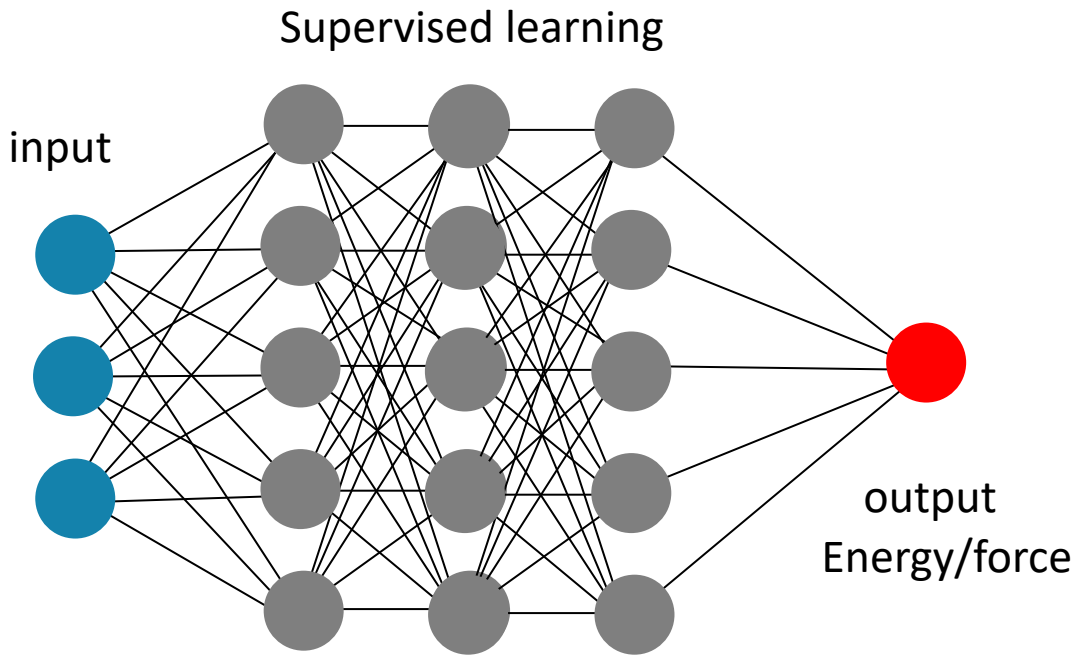
## 第一原理計算は精度が良いけど重すぎる

第一原理計算がしていること　$F: R^N \rightarrow R(energy), R^3(force)$

構造をインプットにしてエネルギーや力を返す

この機能を機械学習モデルで代用できないか？

ポテンシャルを機械学習で作ろう



# 経験的ポテンシャルとの違い

経験的ポテンシャル

物理的な要請から決めた関数形がある

**原子-原子間の相互作用でだいたい決まるだろう**

Embedded Atom Method (EAM) ポテンシャル

$$E_{tot} = \sum_i E_i(n_i) + \frac{1}{2}\sum_{i \neq j} \phi_{ij}(r_{ij})$$

各原子に割り当てられたエネルギー

距離に依存する関数

$$\exp\left[-b^l\left(\frac{r_{ij}}{r_e} - 1\right)\right]$$

などを使うことが多い

**ボンド間の角度も重要なのでは？**

Tersoff ポテンシャル

$$V = \frac{1}{2}\sum_i \sum_{i \neq \mathrm{j}} f_c(r_{ij})[F_R(r_{ij}) + b_{ij}F_A(r_{ij})]$$

$$b_{ij} = \left(1 + \beta_i^n \zeta_{ij}^n\right)^{-1/2n} \qquad \zeta_{ij}^n = \sum_{k \neq i,j} f_c(r_{ik}) g(\theta_{ijk})$$

3つの原子の座標で決まる角度に依存
→多体ポテンシャル

# 経験的ポテンシャルとの違い

経験的ポテンシャル：意味合いもわかりやすい＆解析微分式が得られる→力の計算、MDでの利点

・**パラメータ最適化の労力がとてつもない**

・**パラメータや関数形が固定されてしまう**
→特定の環境に合わせて最適化したパラメータを、他の系で使っていいのかは微妙

Ex) sp2/sp3のように結合にバリエーションがある炭素原子

ダイヤモンド構造はTersoffポテンシャルでも再現できるが、
そのポテンシャルだとグラフェン・カーボンナノチューブでは微妙…

より柔軟性が高く、最適化にも人力を使わなくて済むのが
機械学習ポテンシャルのメリット

# 機械学習ポテンシャルの種類

ポテンシャルの関数形は規定しない

Gaussian Approximation Potential (GAP)

A. P. Bartok et al., Phys. Rev. Lett. 104, 136403 (2010)

カーネル法がベース

(High-dimensional) Neural Network potential (HDNNP)

J. Behler and M. Parrinello, Phys. Rev. Lett. 98, 146101 (2007)

ニューラルネットワーク

ポテンシャルの関数形はある程度決める

Physically informed artificial neural network

EAMや多体ポテンシャルのパラメータを
ニューラルネットワークにする

G. P. Purja Pun et al., Nat. Commun. 10, 2339 (2019)

他の新しいアーキテクチャ：
グラフニューラルネットワークベース

SchNet

K. T. Schutt et al., Arxiv: 1706.08566 (2017)

PFP

S. Takemoto et al., Nat. Commun. 21,2991 (2022)

など

# 機械学習「ポテンシャル」にするには？

## 構造からエネルギーを予測するモデル

構造のインプットがデカルト座標そのままというのはあまり良くない

### 物理的要請

系の持つエネルギーは、空間並進・回転や同種原子の入れ替えでも不変

原子の持つ座標をそのまま入力してしまうと、この対称性を学習させることが難しい

### 学習モデルとしての要請

原子数が異なる系に対応できるモデルである必要がある

# 原子座標をベクトルデータに埋め込む方法

smooth overlap atomic positions (SOAP)

１．原子座標を中心としたガウス関数を設定

２．同径分布関数＋球面調和関数の線形結合で記述し直すと、
原子の重なりは

線形結合の係数

$$p(r_i)_{nn'l}^{Z_1Z_2} = \pi\sqrt{\frac{8}{2l+1}}\sum_m c_{nlm}^{Z_1}(r_i)^* c_{n'lm}^{Z_2}(r_i)$$

のモーメント（$p(r_i)_{nn'l}^{Z_1Z_2} \cdot p(r_j)_{nn'l}^{Z_1Z_2}$）で表現できる

lやmを何次まで取るか？→機械学習モデルの内部パラメータ以外のパラメータ
（ハイパーパラメータ）

S. De et al., Phys. Chem. Chem. Phys. 18, 13745-13769 (2016)

A. P. Bartók, R. Kondor and G. Csányi, Phys. Rev. B, 87, 184115 (2013)

# 原子座標をベクトルデータに埋め込む方法

Atom centered symmetry function（ACSF）

J. Behler, J. Chem. Phys. 134, 074106 (2011)

2つの原子対の距離の分布

$$G_i^2 = \sum_{j \neq i} e^{-\eta\left(R_{ij}-R_s\right)^2} f_c(R_{ij})$$

3つの原子が作る角度の分布

$$G_i^4 = 2^{1-\zeta} \sum_{j \neq i, k \neq i, j} (1 + \lambda \cos\theta_{ijk}) e^{-\eta\left(R_{ij}^2+R_{ik}^2\right)} \; f_c\left(R_{ij}\right) \cdot f_c(R_{ik})$$

$R_s, \eta, \zeta, \lambda$ → ハイパーパラメータ

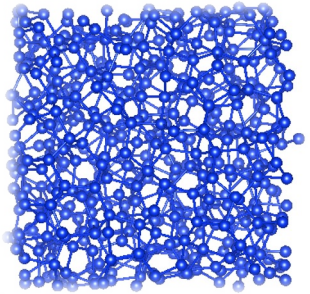
代表的なこれらの記述子の計算
→オープンソースパッケージでも可能　https://singroup.github.io/dscribe/latest/

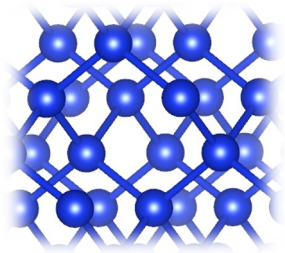
DScribe

1.2.x

# HDNNPの概要

1．様々な構造の第一原理計算結果を集める





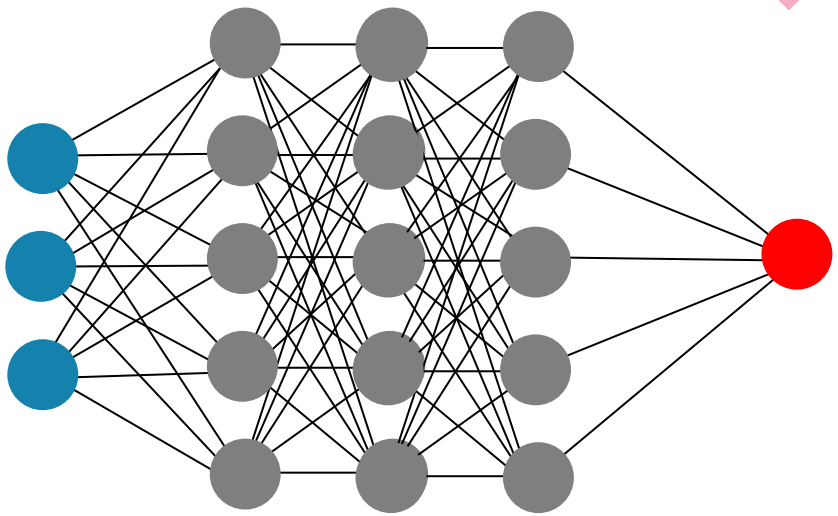
構造　＆　エネルギー・力のペア

2．対称性関数などでベクトルデータに加工

$[G_i^2(\eta_1, \mathrm{R}_{s_1}), \ldots G_i^4(\zeta_1, \eta_1, \lambda_1)]$

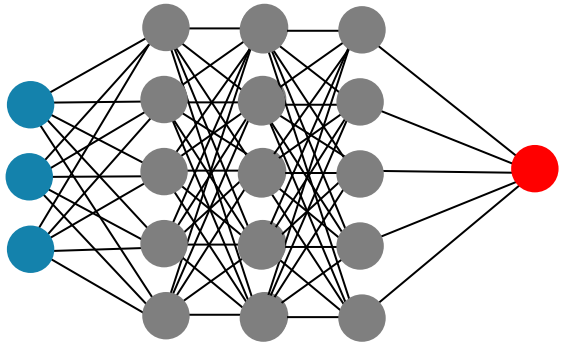
ハイパーパラメータを振ることで
ベクトルになったデータを作る

3．機械学習で重みとバイアスを最適化



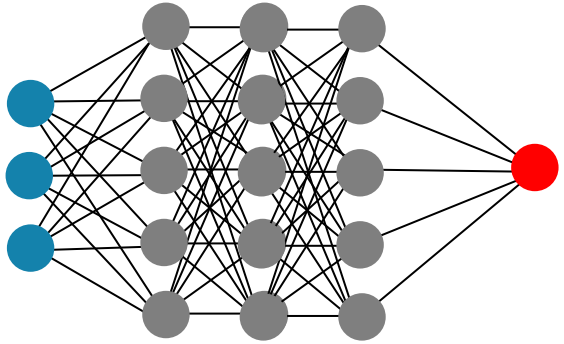
# 何が高次元なのか

原子１の
データ



原子１の
エネルギー

原子２の
データ



原子２の
エネルギー

⋮　⋮　⋮

・ネットワークは同じ種類の元素では同じものを使う
・全体はネットワークの足し上げで表現

→「高次元」

総和→
系の全エネルギー

（この量で損失関数を計算する）

原子数への依存性のなさ
＆同一原子の入れ替えでの不変性を担保

# HDNNPのパッケージ

The *Atomic Energy Network* (**ænet**)

http://ann.atomistic.net/

n2p2

https://compphysvienna.github.io/n2p2/

Simple-NN

https://github.com/MDIL-SNU/SIMPLE-NN_v2

他にも沢山あるが、ある程度ドキュメンテーションが揃っているものを抜粋。

GAPやSchNetベースも含めるとより選択肢が増える。

GAPの場合はQUIPというパッケージが整備されている。https://libatoms.github.io/GAP/index.html
また、VASP6にはGAPの作成機能がある

高度なパッケージは読むのが大変、サクッと概要を眺めてみたい
→トイモデルを用意してあります

https://github.com/eminamitani/sample_NNP

# HDNNPトイモデルを動かしてみる

・手元にファイルをダウンロードしてくる

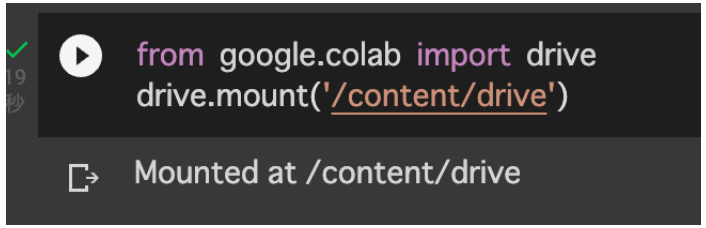
git clone git@github.com:eminamitani/sample_NNP.git　(対称性関数の計算済みデータも含まれている)

・手元にpython3環境あり→仮想環境＋pytorchインストール

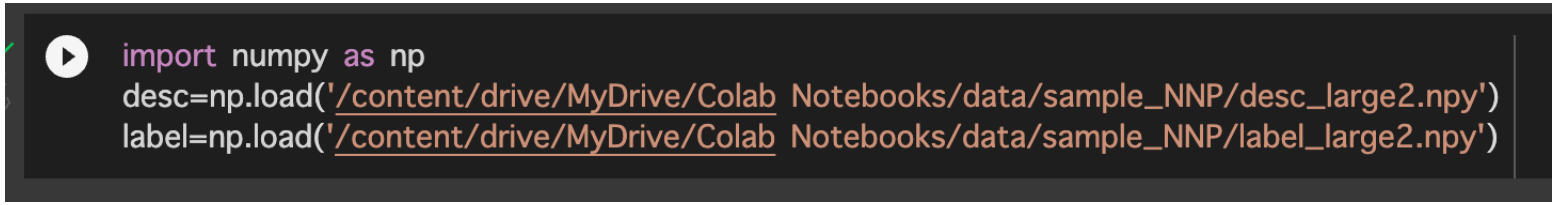
・手元にpython3環境なし→Google Collaboratoryを使う

自分のGoogle Driveにdesc.npy, label.npyをアップロード
sample_NNP.ipynbかsample_NNP2.ipynb(GPU版)をGoogle Collaboratoryでアップロード

冒頭にGoogle Driveのマウント




データの読み込みパスの変更




各自の環境に合わせてください

この２箇所の変更で動くはずです