マテリアルズ・インフォマティクスの基礎と応用

熱物性の第一原理計算

分子科学研究所
大阪大学産業科学研究所

南谷英美

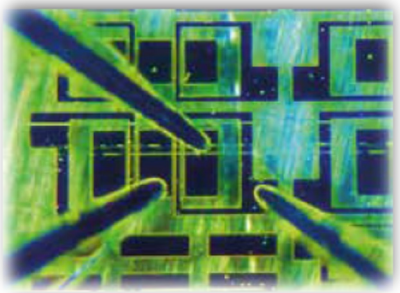
格子物性で応用上重要な量

格子熱伝導率

高熱伝導率

ダイヤモンド：~2000 W/m.K >> シリコン:~100 W/m.K

https://www.itmedia.co.jp/

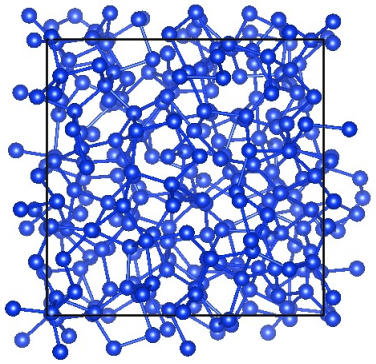


http://nanomari.w3.kanazawa-u.ac.jp/

高効率なパワー半導体

低熱伝導率

アモルファスなど：　アモルファスSi:~1.0 W/m.K



ゼーベック係数

$$Z = \frac{S^2}{\rho\kappa}$$

熱伝導率が低いほど、効率が高い

熱電材料・断熱

熱伝導

熱伝導の基礎法則：フーリエの法則

ジャン・バティスト・
ジョセフ・フーリエ

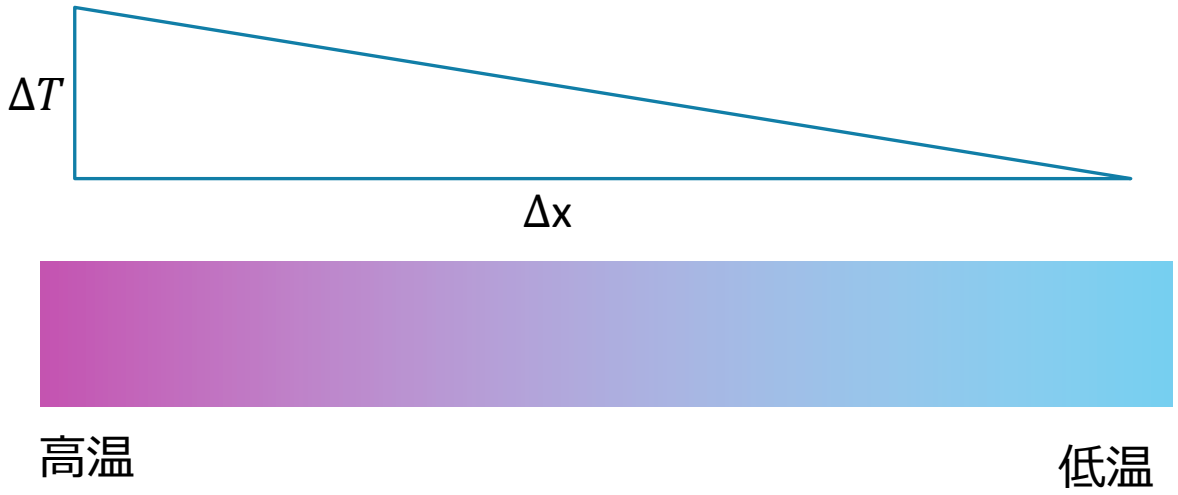
https://ja.wikipedia.org/

熱流束密度　　温度勾配

$$J = -\kappa \frac{\partial T}{\partial x}$$

熱流束密度は温度勾配に比例

比例定数κ：熱伝導率



ミクロな視点での熱伝導

熱の伝搬→フォノンの分布関数の変調に対応しているだろう

ボルツマン方程式

$$\frac{\partial n_q}{\partial t} = \left.\frac{\partial n_q}{\partial t}\right|_{coll} + \left.\frac{\partial n_q}{\partial t}\right|_{diff}$$

温度勾配による拡散

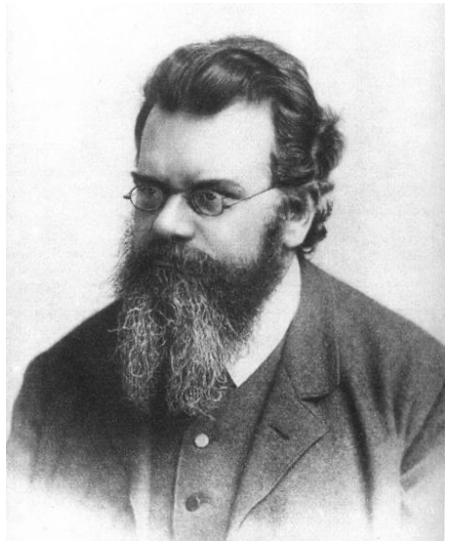
$$\left.\frac{\partial n_q}{\partial t}\right|_{diff} = -\mathrm{v_q}\cdot\nabla T\frac{\partial n_q}{\partial T}$$

フォノンとフォノンの衝突？

調和近似の元ではフォノンはお互い無干渉のはずでは？

非調和効果

原子間のポテンシャルが実は調和振動子のポテンシャルから
ずれていることがフォノンとフォノンの相互作用を生む



ルードヴィッヒ・ボルツマン

熱伝導率の理論式の導出

・定常状態ではフォノンの分布関数は全体としては変わらないはず

$$\left.\frac{\partial n_q}{\partial t}\right|_{coll} = \mathrm{v_q}\cdot\nabla T\frac{\partial n_q}{\partial T} \approx \mathrm{v_q}\cdot\nabla T\frac{\partial n_q^0}{\partial T}$$

拡散項との釣り合い　　平衡状態で近似（線形化）

・あるフォノンモード(波数q)の分布関数の時間変化

平衡状態での分布関数

$$\left.\frac{\partial n_q}{\partial t}\right|_{coll} = -\frac{n_q - n_q^0}{\tau_q}$$

緩和時間近似
（非調和効果による緩和）

$$n_q - n_q^0 = -\tau_q v_\mathrm{q}\cdot\nabla T\frac{\partial n_q^0}{\partial T}$$

・格子気体モデル

$$J = \frac{1}{V}\sum_q v_q\hbar\omega_q(n_q - n_q^0) \longrightarrow J = -\frac{1}{V}\sum_q \tau_q v_q \otimes v_q\hbar\omega_q\frac{\partial n_q^0}{\partial T}\nabla T$$

熱伝導率と非調和効果

ボルツマン方程式＋緩和時間近似＋格子気体モデル

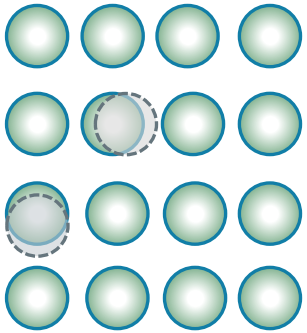
$$J = -\frac{1}{V}\sum_q \tau_q v_q \otimes v_q \hbar\omega_q \frac{\partial n_q^0}{\partial T}\nabla T$$

フーリエの法則

$$J = -\kappa \nabla T$$

$$\kappa = \frac{1}{V}\sum_q \tau_q v_q \otimes v_q \hbar\omega_q \frac{\partial n_q^0}{\partial T}$$

緩和時間に非調和効果が含まれている



τ_q　→３次の力の定数とボーズ分布で決まる

$$\Phi_{\alpha\beta\gamma}(lb, l'b', l''b'') = \frac{\partial^3 V}{\partial u_\alpha(lb)\partial u_\beta(l'b')\partial u_\gamma(l''b'')}\bigg|_0 = -\frac{\partial^2 F_\gamma(l''b'')}{\partial u_\alpha(lb)\partial u_\beta(l'b')}$$

原子２つが様々なパターンで動いたときの力の計算が必要

第一原理計算の精度がほしいけど計算コストが高い→機械学習ポテンシャル

機械学習ポテンシャル

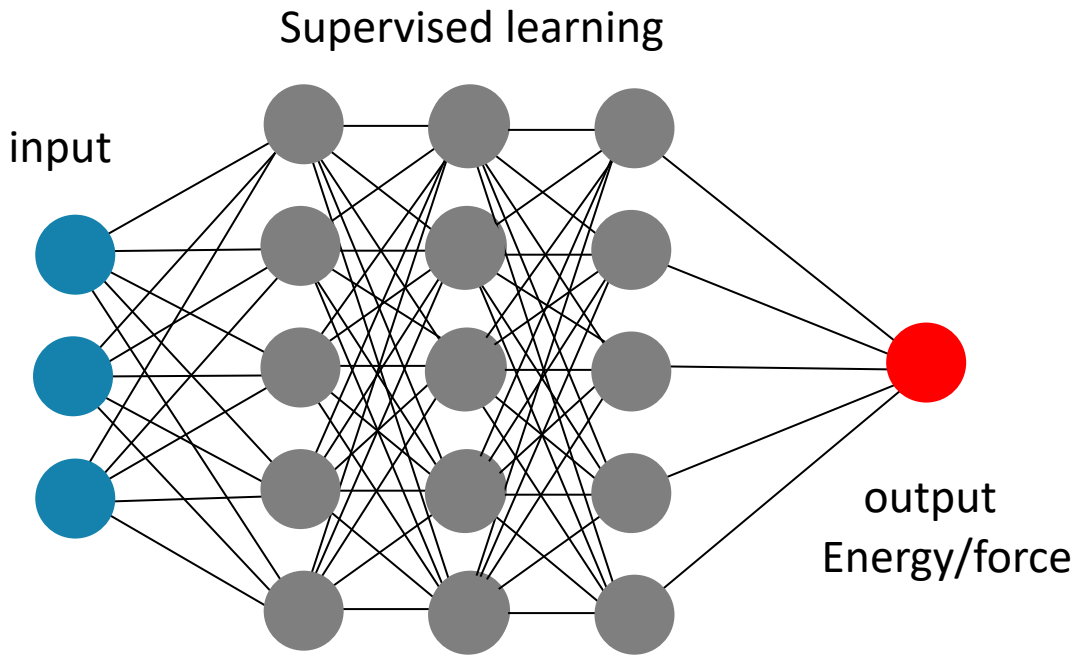
第一原理計算は精度が良いけど重すぎる

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. Bartók 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

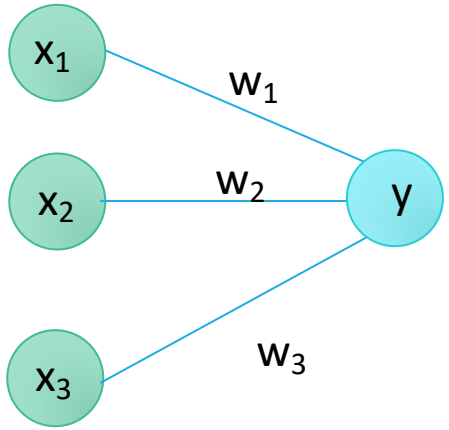
S. Takemoto et al., Nat. Commun. 21,2991 (2022)

など

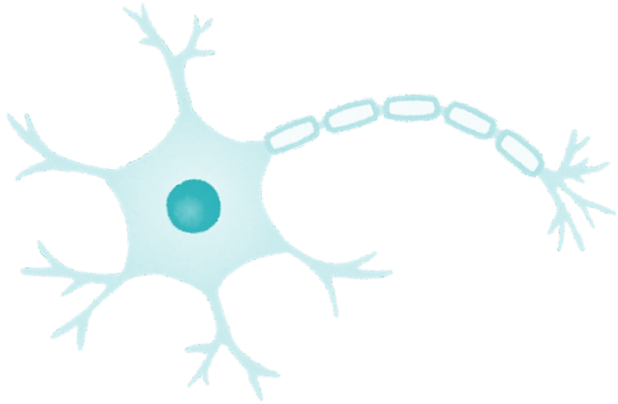
ニューラルネットワーク

生物の神経系・脳→ニューロンの発火の伝搬で情報処理

ニューラルネットワークの構成要素：パーセプトロン ⟷ ニューロンの発火モデル



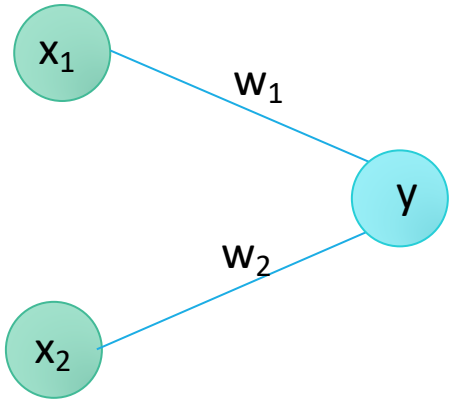
$$y = \begin{cases} 0 & (w_1x_1 + w_2x_2 + w_3x_3 \leq \theta) \\ 1 & (w_1x_1 + w_2x_2 + w_3x_3 > \theta) \end{cases}$$



入力に重みをかけたものの総和が
しきい値より大きいか小さいかで、0または1を返す

参考文献：ゼロから作るDeep Learning　斎藤康毅　著　（オーライリー・ジャパン）

パーセプトロンでできること



ANDゲートの動作

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

$w_1=0.3$, $w_2=0.6$, $\theta=0.8$

$w_1x_1 + w_2x_2$

0

0.6<θ

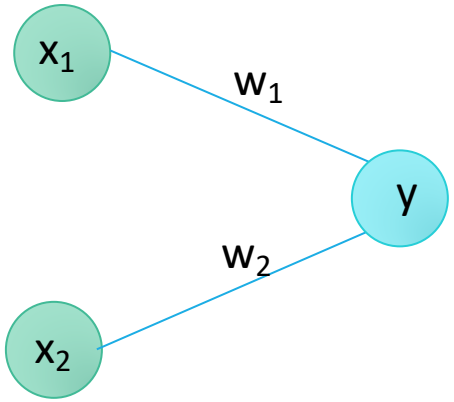
0.3<θ

0.9>θ

他にも様々な組み合わせで実現できる

パーセプトロンでできること

重みやしきい値を変えるだけでOR, NANDゲートの動作にもできる



OR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Ex) w_1=0.3, w_2=0.6, θ=0.2

NAND

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

Ex) w_1=-0.3, w_2=-0.2, θ=-0.4

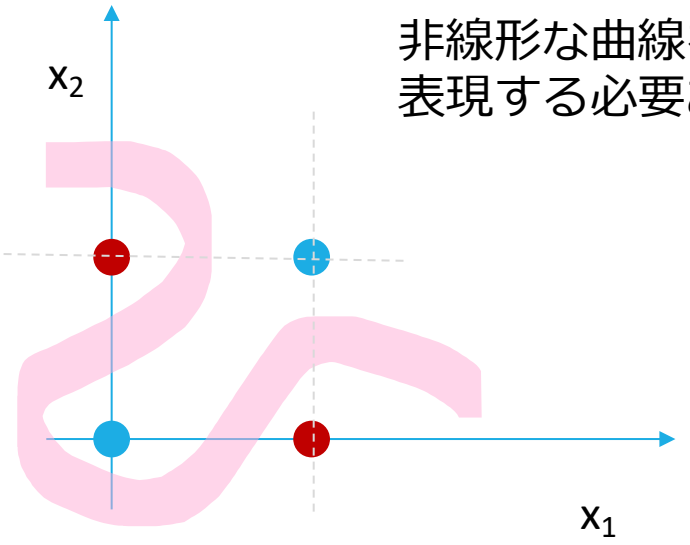
パーセプトロンでできること

XORゲートは作れるか？

XOR

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	1

XOR→線形分離できない



非線形な曲線を
表現する必要あり

パーセプトロン１つでは無理

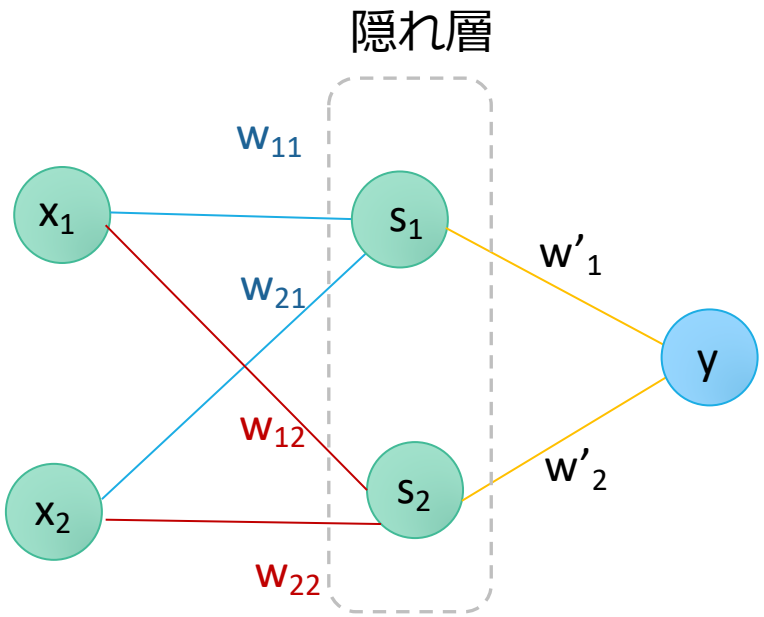
↓

重ねると実現できる

↓

多層パーセプトロン

多層パーセプトロン



x_1	x_2	s_1	s_2	y
0	0	1	0	1
0	1	0	0	0
1	0	0	0	0
1	1	0	1	1

考えられる例

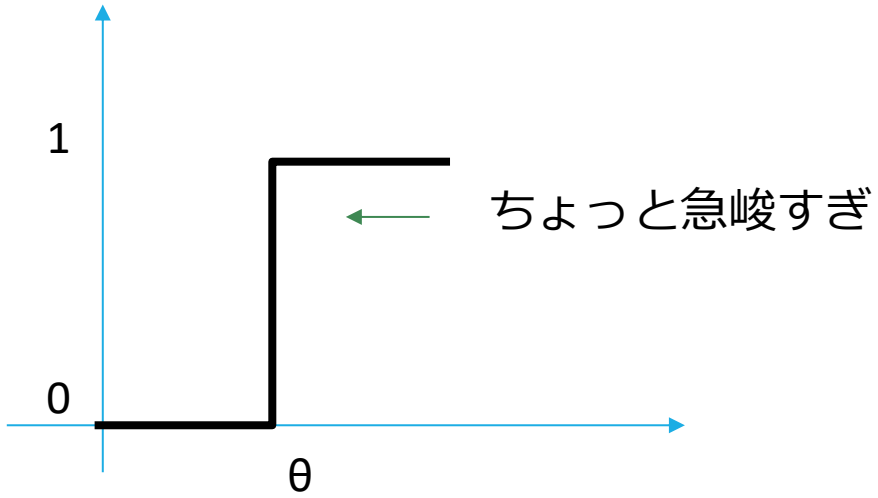
青い部分(w_{11}=-0.5, w_{21}=-0.5, θ=0)
赤い部分(w12=0.5, w22=0.5, θ=0.8)
黄色い部分 (w'1=0.4, w'2=0.6, θ=0.3)

パーセプトロンを多層にすることで非線形な関数も表現できる
→万能近似定理

（隠れ層が一つあればいくつかの条件のもと、
任意の連続関数を近似できる）

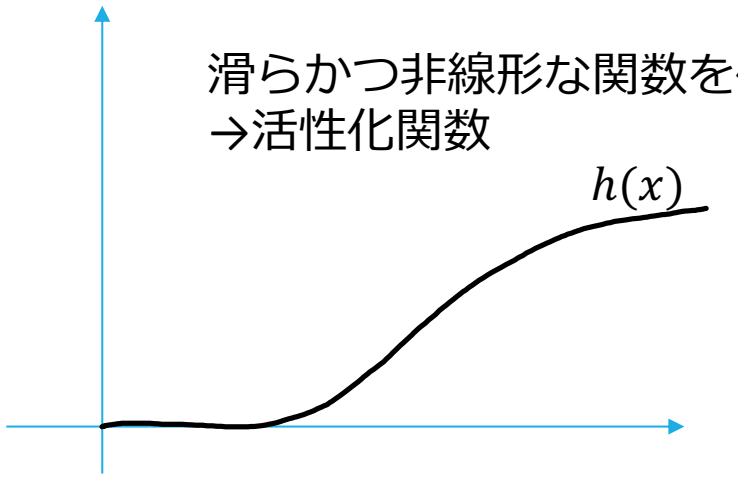
活性化関数

パーセプトロンではある閾値より上か下かで０，１を切り替えていた



$$y = \begin{cases} 0 & (w_1x_1 + w_2x_2 \leq \theta) \\ 1 & (w_1x_1 + w_2x_2 > \theta) \end{cases}$$

滑らかつ非線形な関数を使おう
→活性化関数

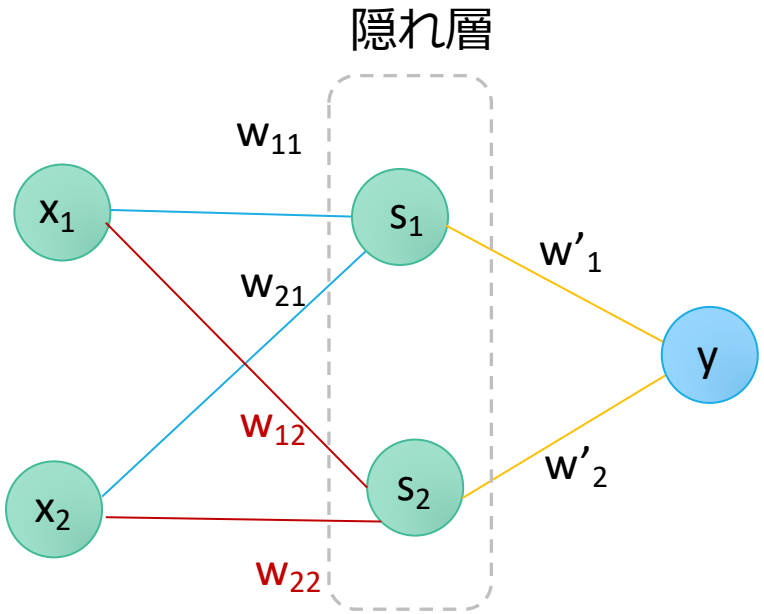


$$y = h(w_1x_1 + w_2x_2)$$

シグモイド関数など　$h(x) = \dfrac{1}{1 + e^{-ax}}$

ニューラルネットワーク&機械学習へ

インプットと出力の関係を実現する重み・閾値（バイアス）は無限にある



活性化関数を滑らかにしておくことで、
微分を使った誤差逆伝播法で重み・バイアスを
最適化できる

あるパラメータのネットワークから得られた出力と、
真の値で決まる誤差：損失関数の計算結果

$Loss(y, y_{true})$　e.g. 最小二乗誤差

その偏微分をつかってパラメータを更新する

$$w_{11} = w_{11} - \eta \frac{\delta(Loss(y, y_{true}))}{\delta w_{11}}$$

勾配法

これを繰り返すことでよい近似ネットワークを得る→機械学習

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)

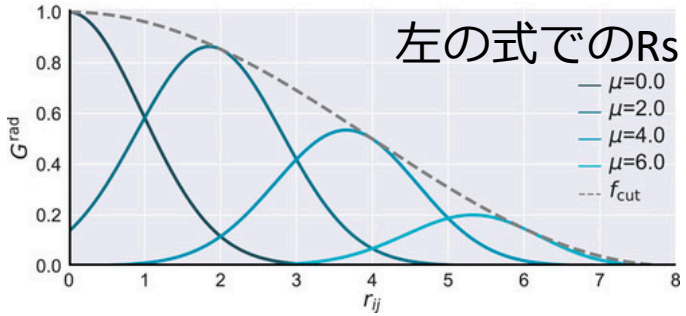
原子座標をベクトルデータに埋め込む方法

Atom-centered symmetry function （ACSF）

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

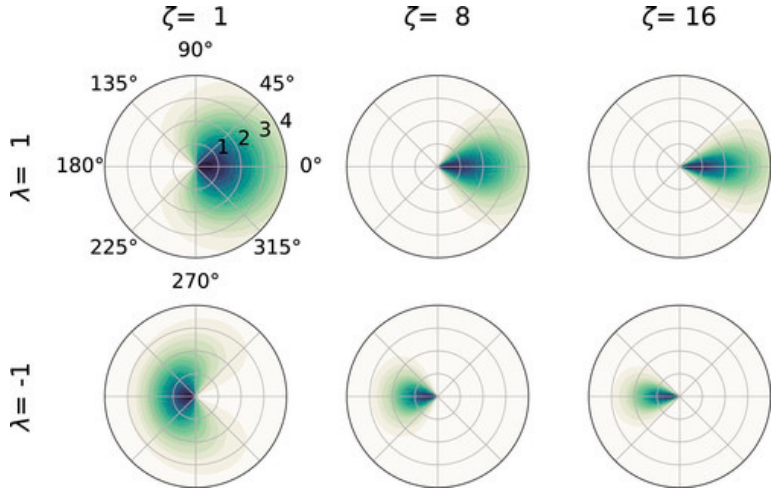
2つの原子対の距離の分布

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



3つの原子が作る角度の分布

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



M. Gastegger et al., J. Chem. Phys. 148, 241709 (2018)

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

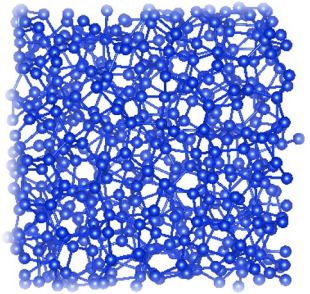
代表的なこれらの記述子の計算→オープンソースパッケージでも可能
https://singroup.github.io/dscribe/latest/

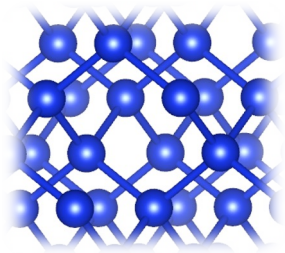
DScribe

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





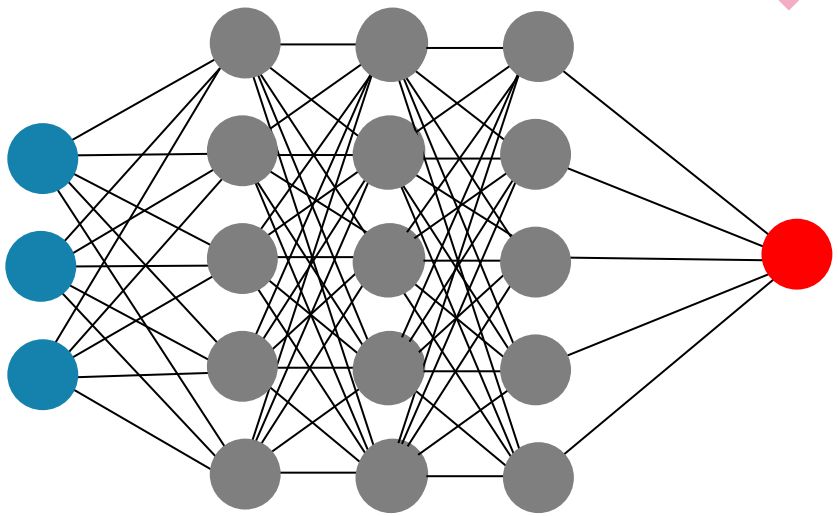
構造　＆　エネルギー・力のペア

2．対称性関数などでベクトルデータに加工

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

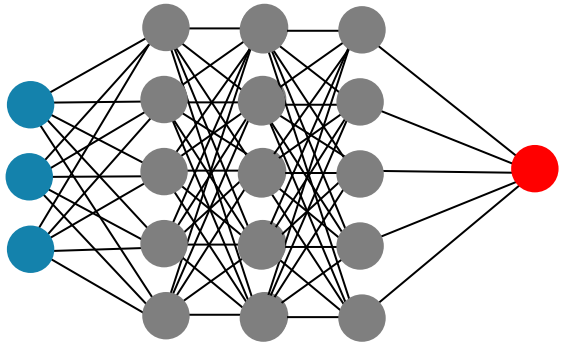
ハイパーパラメータを振ることで
ベクトルになったデータを作る

3．機械学習で重みとバイアスを最適化



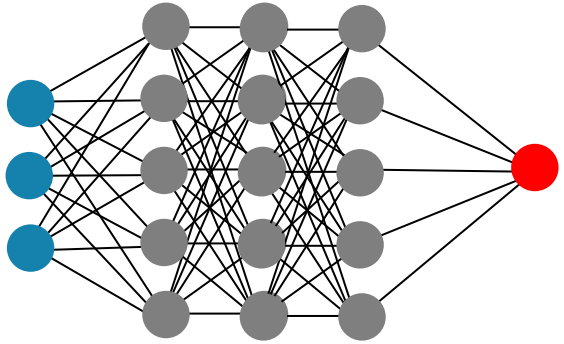
何が高次元なのか

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

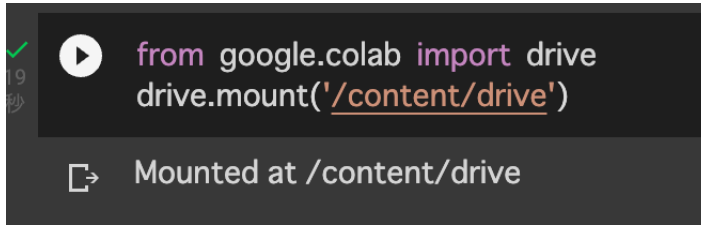
git clone git@github.com:eminamitani/sample_NNP.git　(対称性関数の計算済みデータも含まれている)

・手元にpython3環境あり→仮想環境＋pytorchインストール

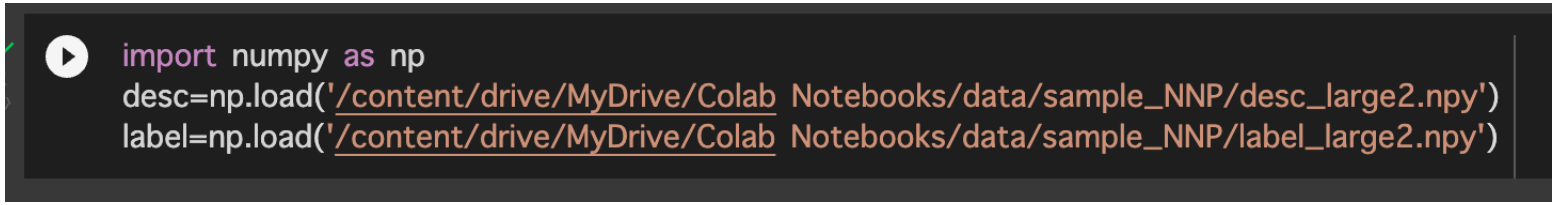
・手元にpython3環境なし→Google Collaboratoryを使う

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

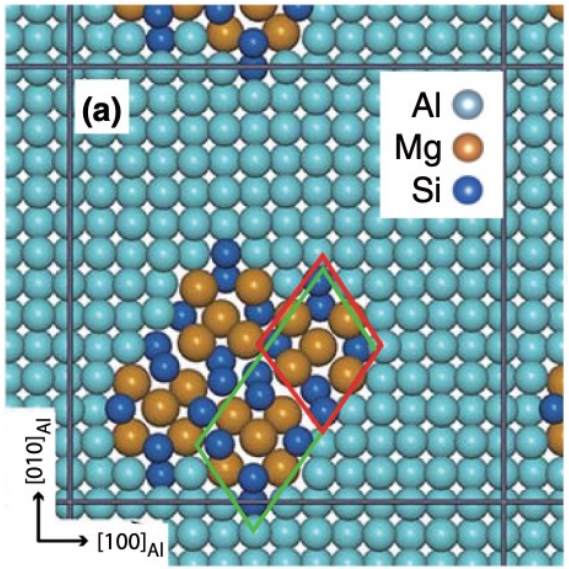


各自の環境に合わせてください

この２箇所の変更で動くはずです

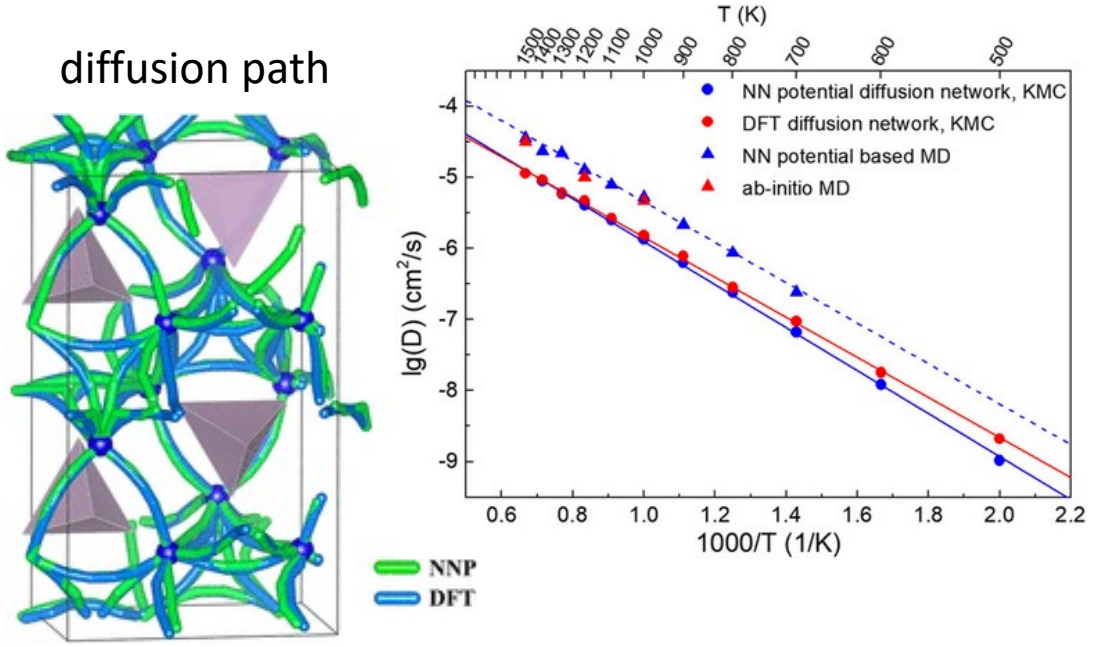
HDNNPの応用例

合金の形成エネルギー
→実現される組成・構造の予測



R. Kobayashi et al.,
Phys. Rev. Mater. 1, 053604 (2017).

アモルファス中のイオン伝導

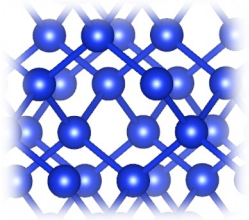


W. Li et al., J. Chem. Phys. 147, 214106 (2017).

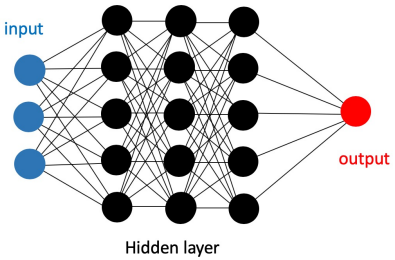
他にも多数　　様々な構造でのエネルギーや力の情報が必要とされるタスクに有用

熱伝導率への応用

平衡MDで様々な変位
を含んだ構造を作成

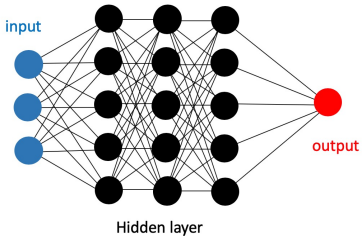


HDNNPを訓練



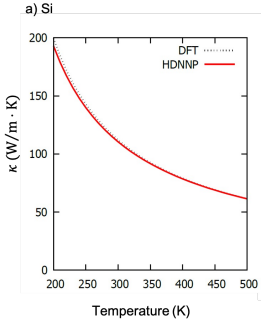
phono3pyを使って
必要な変位パターンを決定

各パターンでの力を決定



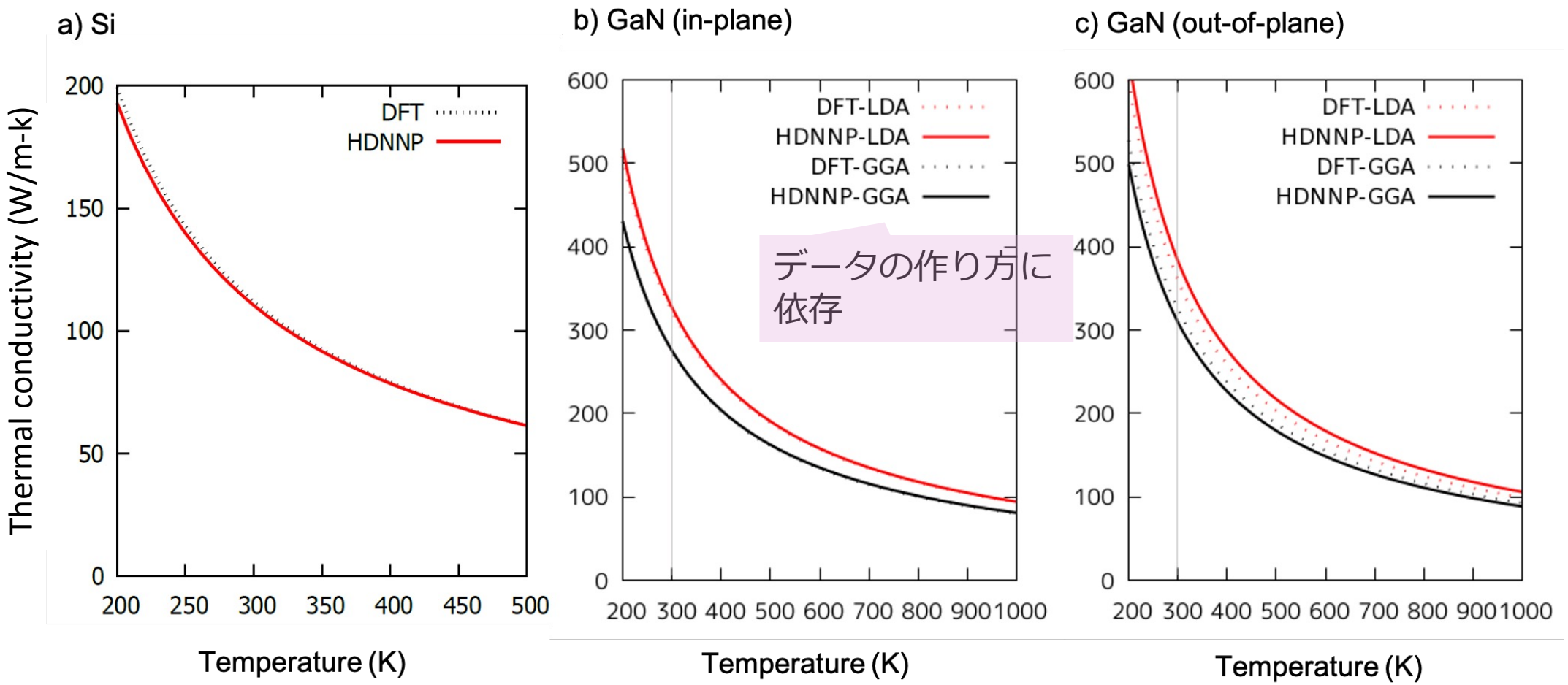
力の定数や緩和時間の計算

熱伝導率



熱伝導率への応用

APEX 12, 095001 (2019)



第一原理計算に匹敵する精度を1/800の時間で実現

（データを作る時間が課題）

後半のまとめ

・熱伝導率の理論・計算のボトルネック

・機械学習ポテンシャルとニューラルネットワーク

・機械学習ポテンシャルの応用