神戸大学
計算生命科学

1.5
「AI 創薬：創薬における人工知能と機械学習の基礎および応用」

《11月2日（水）》

舘野 賢
日本たばこ産業株式会社
医薬総合研究所
主席研究員

人工知能（AI）による創薬とは？

AI が飛躍的に発展している！

囲碁、将棋、etc.

AlfaGo → 囲碁のチャンピオンを破った！

画像処理（Image Processing）の技術

人を見分ける能力が、人の能力を超えた！

自然言語処理（Natural Language Processing）

異なる言語の間の翻訳だけではなく、小説を書く！

簡単な四則演算もする！　画像処理やその他のタスクにも応用！

創薬に AI が活用できるか？　→ できる！

AlphaFold2 → Deep Learning により、

タンパク質の立体構造の予測が可能になった！？

人工知能（AI）による創薬とは？

創薬に AI が活用できるか？　→ できる！

他方で、AI の Fields における日本の Presence は？？？

そこで・・・

あまり見慣れない（難しい！？）Concepts なども、先端的で重要な内容については
やさしく（！）お話しすることに挑戦したいと思います
ただしその分、直感的なお話しになるため、数学的により Strict な内容は、
　テキストや論文などを御覧下さい。
（例えば、AI の内部がどのように動くのかも、できるだけお話してみたいと思います）

また、**以降のスライドの中の日本語や英語の文章は（講義の際にはあまり読まずに、復習などのために！）、**ポイントした Terms と Figures, Tables などを中心に御覧下さい、
できる限りそのような形で理解が進むように、お話したいと思います。

人工知能（AI）による創薬とは？

創薬のあらゆる領域において、 AI が既に活用されている！

どんな Fields で？

医薬品の**分子設計**など ← Chemistry and Biophysics

医薬品の Physical, Physico-Chemical, Chemical, and Biological Properties

化合物の合成技術

医薬品の**薬理効果**や**体内動態**など ← Biochemistry and Biology

化合物の薬効、安全性（Toxicity）、

化合物の Pharmaceutical Kinetics and Dynamics (PK and PD)

医薬品開発 **Target** の **Identification** ← Biology and Genome Science

Transcriptome, Proteome, Epigenetics, Interactome, etc.

Protein-Protein Interaction (PPI)

医薬品の**生産技術**など ← Engineering

製造プロセス技術、製剤技術、(結晶) 構造解析技術、品質管理、etc.

人工知能（AI）による創薬とは？

創薬のあらゆる領域において、 AI が既に活用されている！

今日の Topics は？

分子設計（I）

タンパク質の**アミノ酸配列**から、結合サイトとリガンドを Identify（Affinity）

分子設計（II）

タンパク質・化合物の複合体の **3D Structural Docking** （& Affinity）

医薬品の**電子構造**

薬理効果、PK and PD、安全性、分子設計（III）、etc.

→ 物質の様々な性質（Properties）や生体反応（Reactivity）を理解するための基盤

→ Screening and Reactivity などの探索における基盤

創薬のためのターゲットの探索　→ Genome Science の応用も！

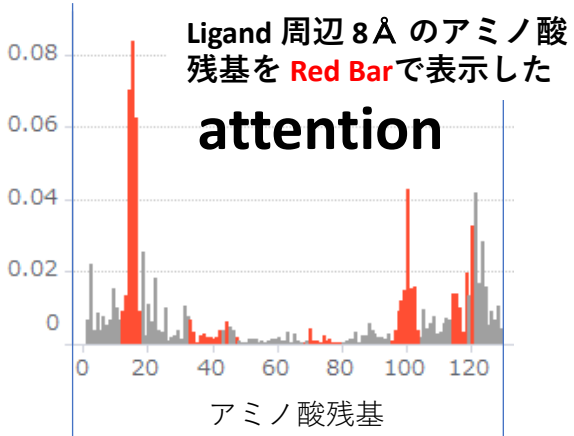
Transcriptome, **Proteome**, **Interactome**, **Epigenetics**, etc.

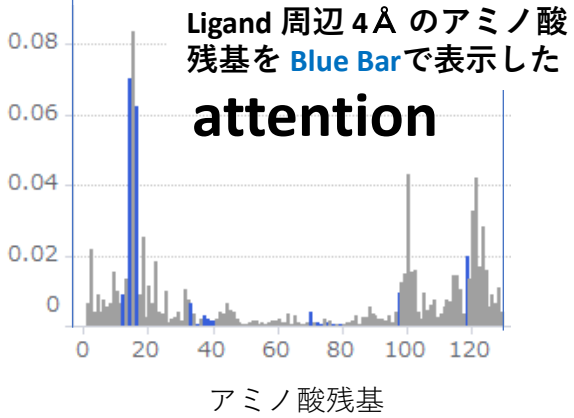
→ **Multi Omics** Analysis

<Attention Values of Amino Acid Residues and Ligand Atoms>

アミノ酸残基および化合物の相互作用部位を学習するか？

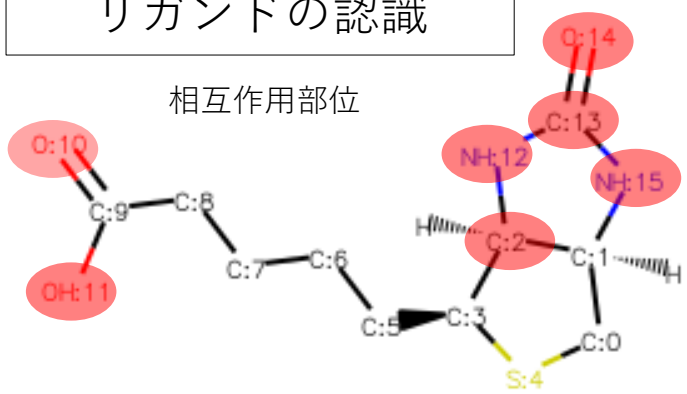
アミノ酸残基の認識

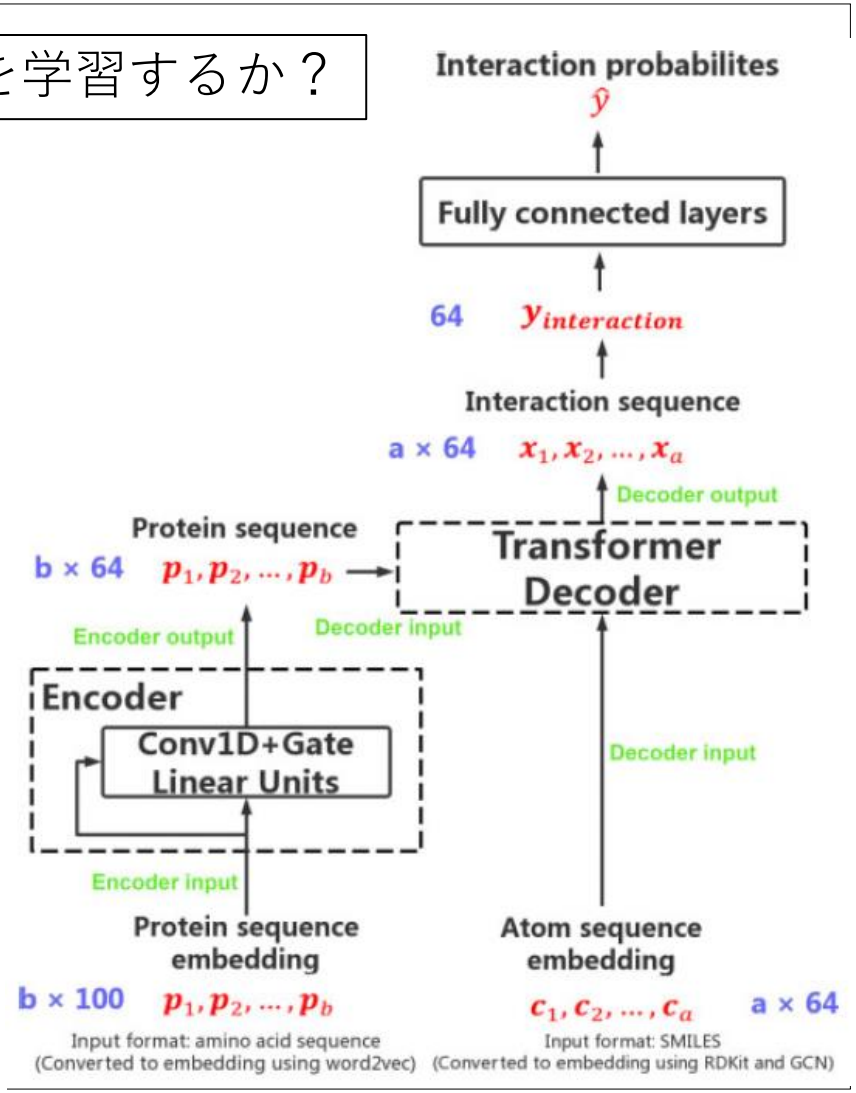




解析のイメージ

リガンドの認識





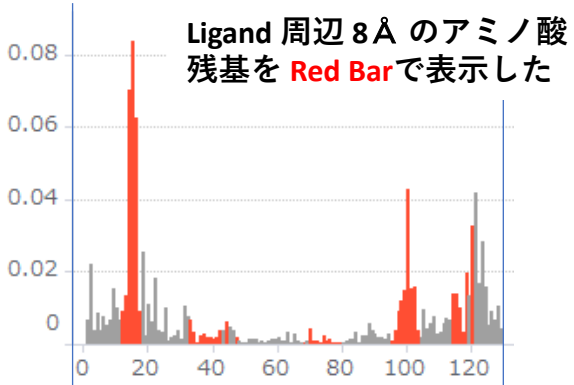
アミノ酸残基をベクトル化

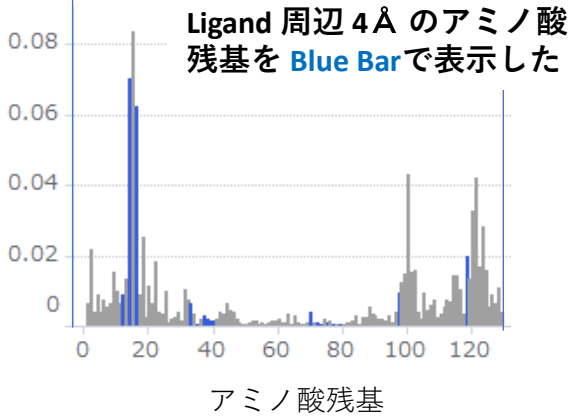
化合物の原子をベクトル化

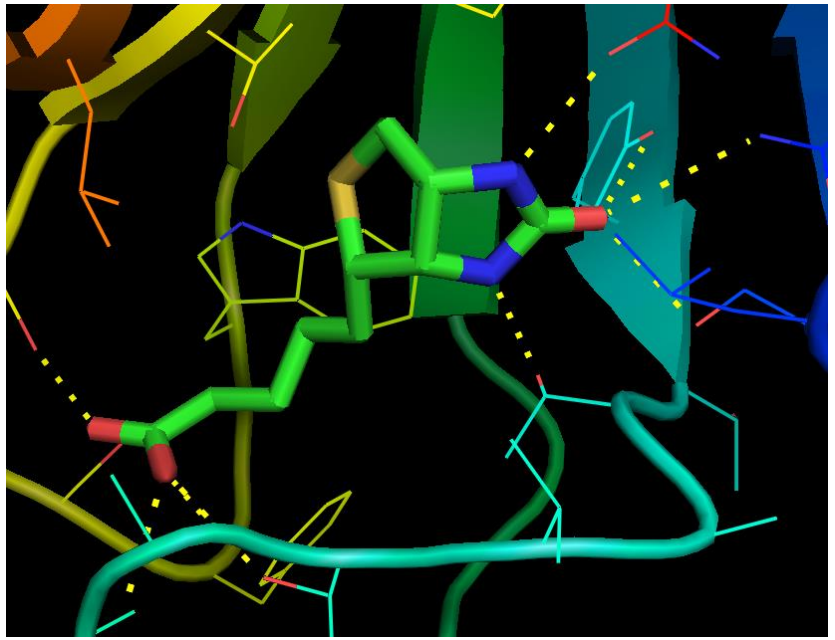
<Attention Values of Amino Acid Residues and Ligand Atoms>

アミノ酸残基および化合物の相互作用部位を学習するか？

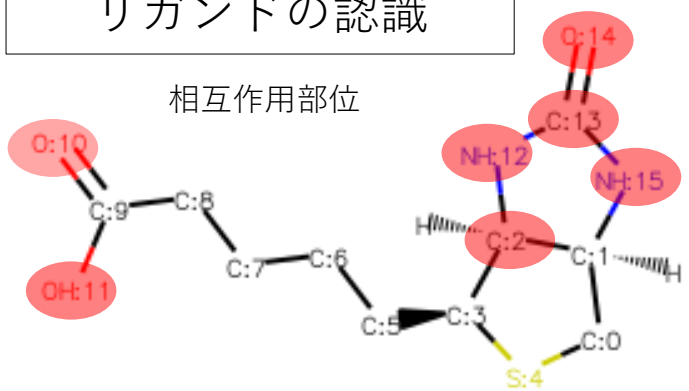
アミノ酸残基の認識

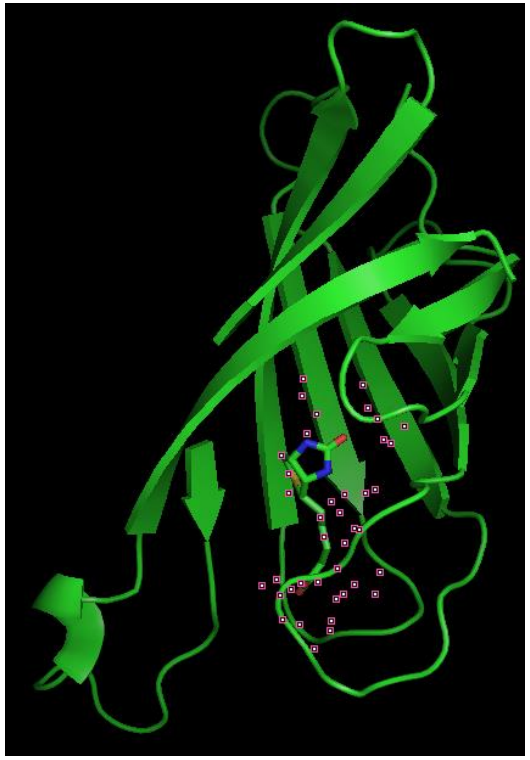






リガンドの認識





Avidin-Biotin (2AVI)

人工知能（AI）による創薬とは？

創薬のあらゆる領域において、 AI が既に活用されている！

今日の Topics は？

AI 創薬を理解するために・・・

AI は何をしている？

・どんなシステムが使われている？　そのシステムで・・・

・どんな計算をしているのかな？

・何ができるようになったかな？

・どんな課題があるのか？

AI を学ぶために、どんな勉強をすれば良いかなあ？

AI を創るために、どんな **Tactics** が大切かなあ？

今日のお話しの基本的な方針を再度、確認して・・・

Brief History of Artificial Mathematical Neural Network

Beginning of Mathematical Models in Neural Network Fields (1943)

Artificial Neuron

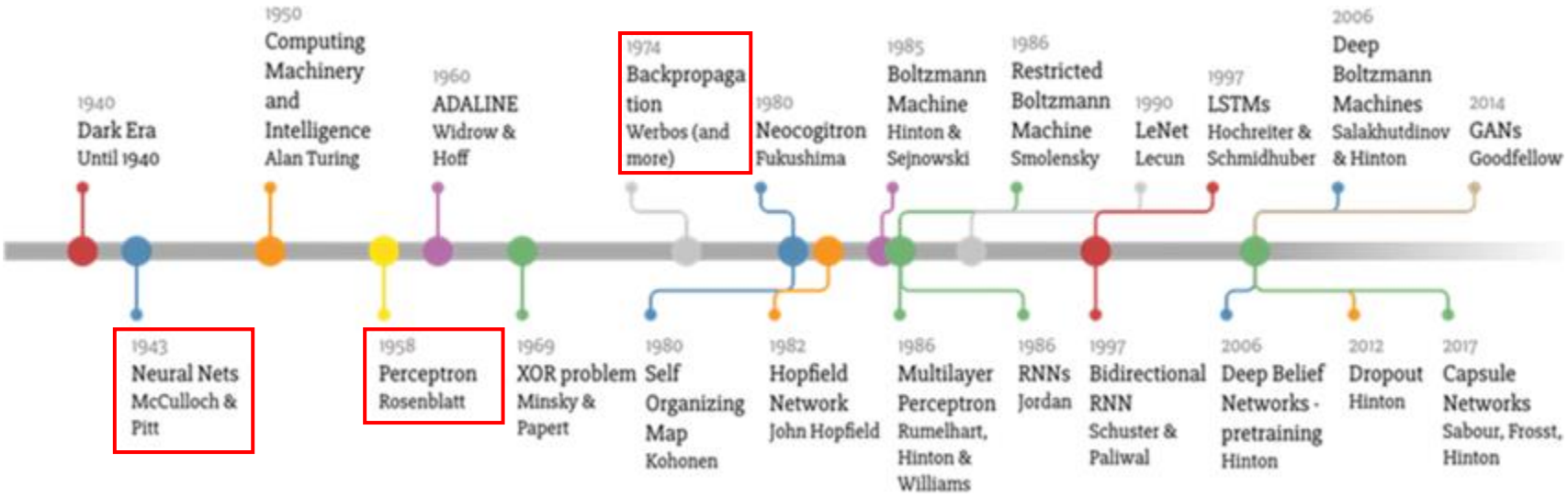
Mathematical Model of Neuron Proposed by McCulloch and Pitts

McCulloch, W. : neurophysiologist

Pitts, W. : mathematician

1st Era of Artificial Neural Network Model (1950's to 1960's)

Perceptron : a Feed-Forward Neural Network Model, Proposed by Roseblatt, F. in 1956



D. Mourtzis and J. Angelopoulos, *Int. J. Adv. Manuf. Syst.* (2020)

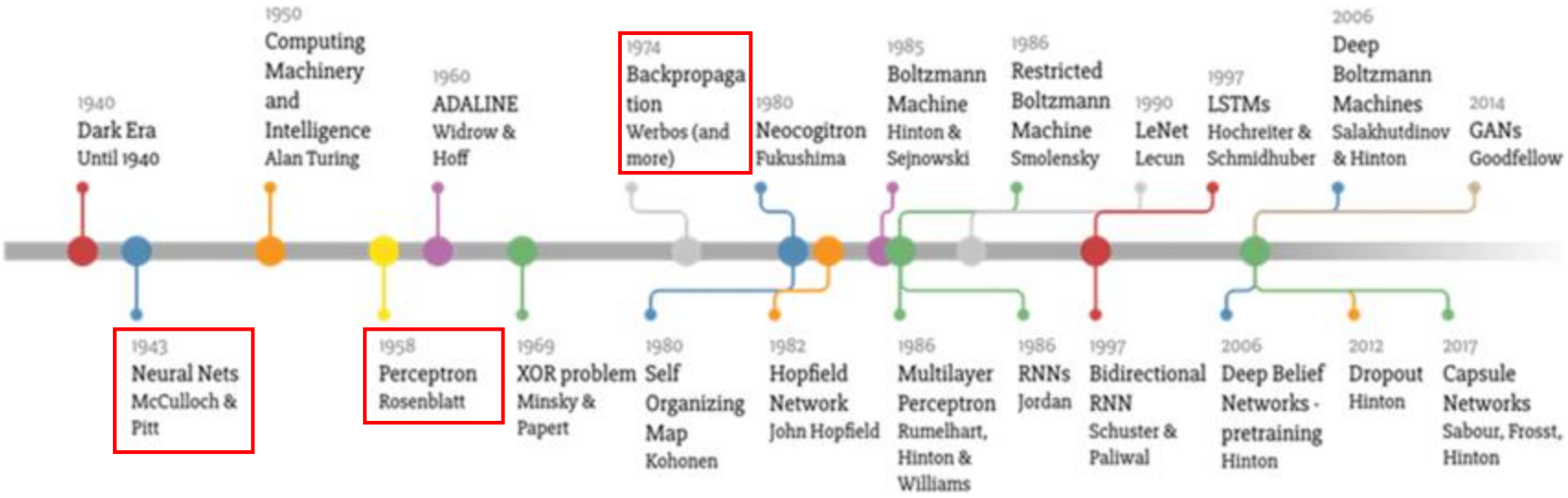
Brief History of Artificial Mathematical Neural Network

2nd Era of Artificial Neural Network Model (1980's to 90's)

Feed Forward Multilayer Network : Learning (Training) by Back Propagation
Hopfield Networks, Boltzmann Machine, Cognitron (1975)
Cf)
"*Expert System*" : a System for Exploring the Accumulated Knowledge in Relevant, Specific Fields

3rd Era of Artificial Neural Network Model (2010's to the present days)

"*Deep Learning*" : a Trigger of the Beginning of this new Era.

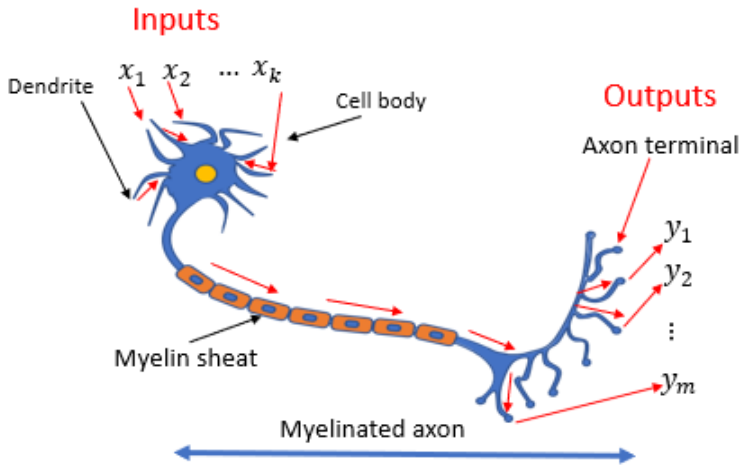


D. Mourtzis and J. Angelopoulos, *Int. J. Adv. Manuf. Syst.* (2020)

Brief History of Artificial Mathematical Neural Network

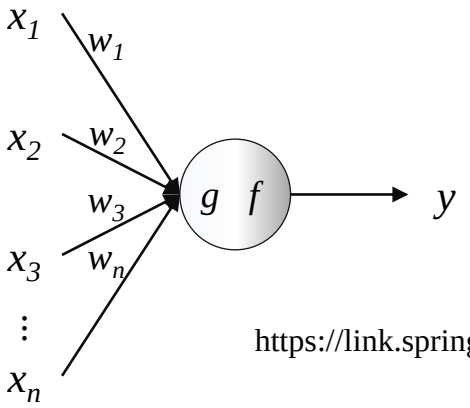
Neuron Model Proposed by McCulloch-Pitts (1943)

Biological Neuron



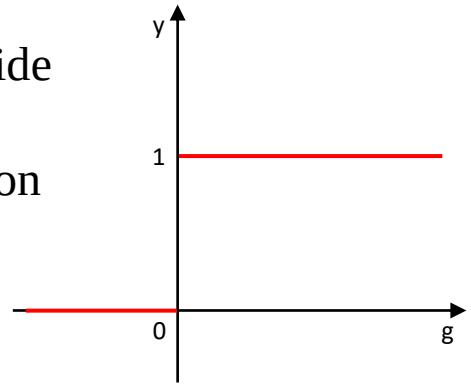
T. Fukuchi et al.,

McCulloch-Pitts Neuron Model (1943)



https://link.springer.com/article/10.1007/BF02478259

Heaviside Step Function



https://www.sciencedirect.com/topics/mathematics/heaviside-unit-step

Summation of inputs

$$g = \sum_{i=1}^{n} x_i w_i$$

Activation Function

$$y = f_{(g)} = \begin{cases} 1 \ (g > 0) \\ 0 \ (g \leq 0) \end{cases}$$

Perceptron-Type Neural Network

Activation Functions and their Derivatives

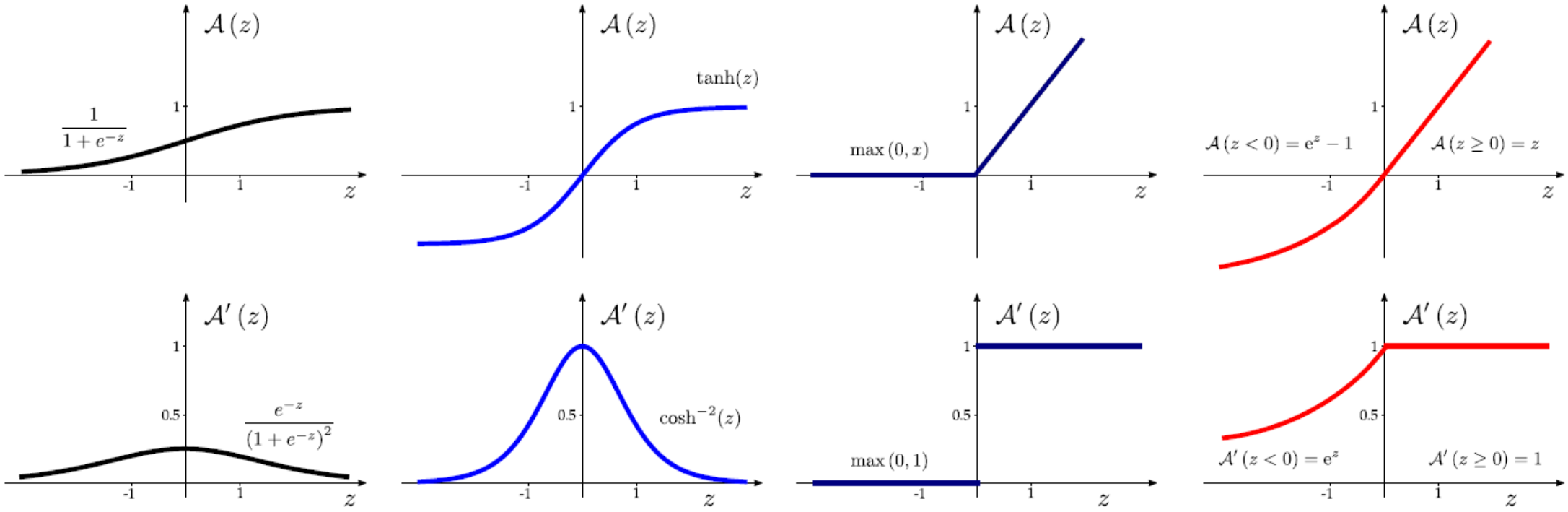


Figure 4: Some of the most common activation functions and their first-order gradient. From left to right: the logistic (sigmoid) function, the hyperbolic tangent, the Rectified Linear Unit (ReLU), and the Exponential Linear Unit (ELU).

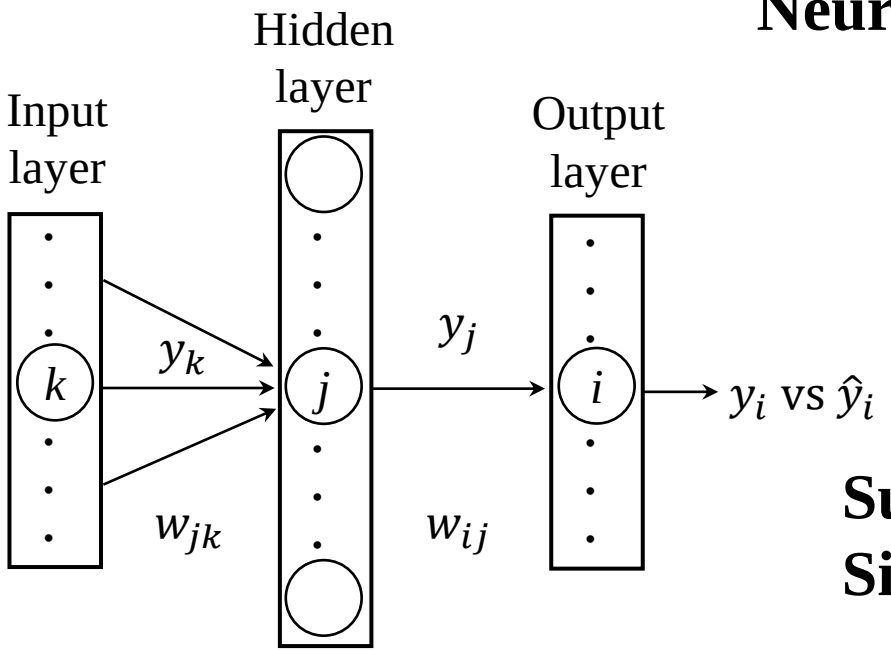
https://www.researchgate.net/publication/346898697_Thermodynamics-based_Artificial_Neural_Networks_for_constitutive_modeling

Backpropagation

Multi Layer Perceptron-Type Neural Network

Perceptron (3 layers)

$$\sum_k w_{jk} x_k = x_j$$
$$y_j = \sigma(x_j)$$
$$x_i = \sum_j w_{ij} y_j$$
$$y_i = \sigma(x_i)$$

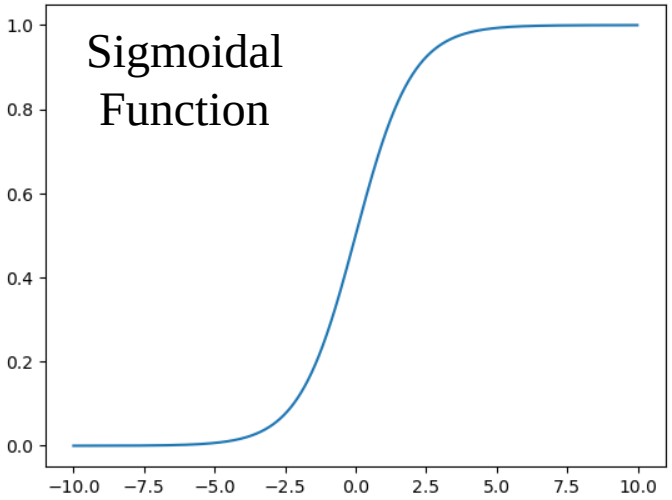


Supervise Signals

Activation Function

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\sigma'(z) = \frac{d\sigma}{dz}$$
$$= \frac{e^{-z}}{(1 + e^{-z})^2} = \frac{1}{1 + e^{-z}} \times \left(1 - \frac{1}{1 + e^{-z}}\right)$$
$$= \sigma(z)(1 - \sigma(z))$$

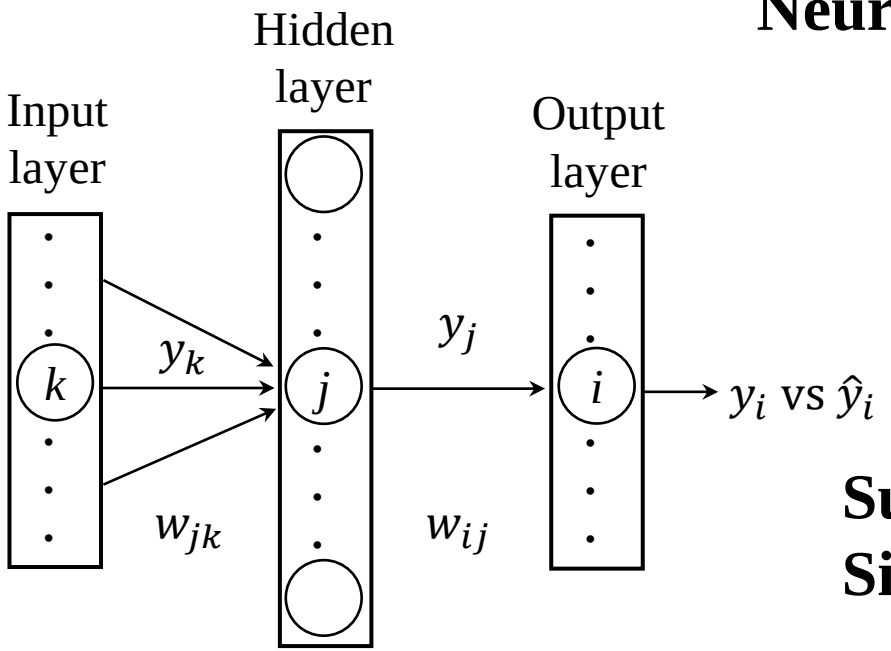


Backpropagation

Multi Layer Perceptron-Type Neural Network

Perceptron (3 layers)

$$\sum_k w_{jk} x_k = x_j$$
$$y_j = \sigma(x_j)$$
$$x_i = \sum_j w_{ij} y_j$$
$$y_i = \sigma(x_i)$$



Supervise Signals

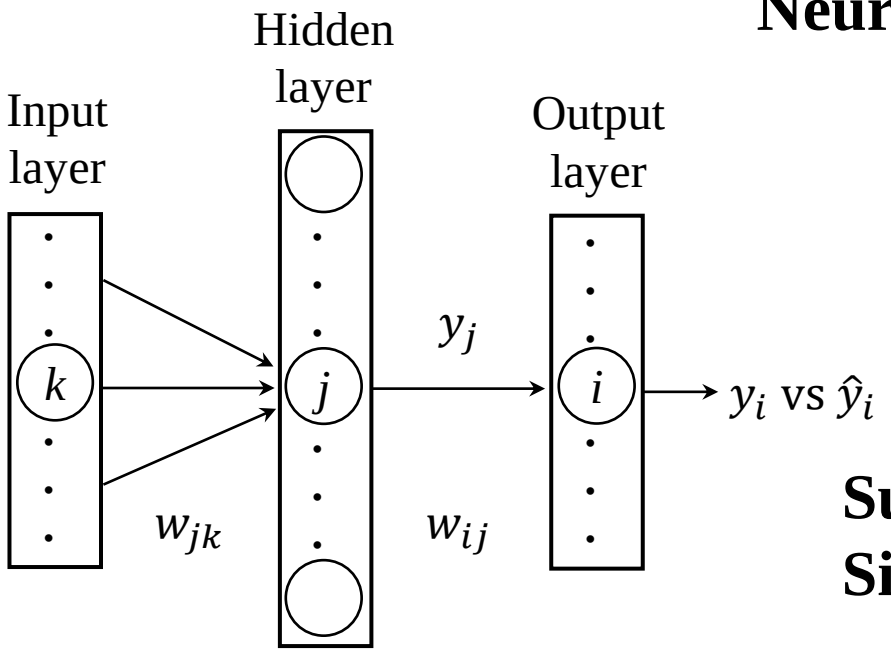
$$W_1 = \begin{pmatrix} w_{jk} \end{pmatrix} \qquad W_2 = \begin{pmatrix} w_{ij} \end{pmatrix}$$

$$x_j = \sum_j w_{jk} y_k = W_1 \mathbf{y}^{(j)} \qquad x_i = \sum_j w_{ij} y_j = W_2 \mathbf{y}^{(i)}$$

Products of Matrices and Vectors

Backpropagation

Multi Layer Perceptron-Type Neural Network



Supervise Signals

Error Function / Loss Function

$$E = \frac{1}{2}\sum_i (y_i - \hat{y}_i)^2$$

$$\frac{dE}{dt} = \sum_i \frac{\partial E}{\partial w_{ij}} \frac{dw_{ij}}{dt}$$

とおく。ここで,

$$\frac{dw_{ij}}{dt} = -\frac{\partial E}{\partial w_{ij}} \cdots\cdots (*)$$

とおくと

$$\frac{dE}{dt} = -\sum_i \left(\frac{\partial E}{\partial w_{ij}}\right)^2 \leq 0$$

となり，Eは増えない

$$\nabla f(\boldsymbol{r})$$

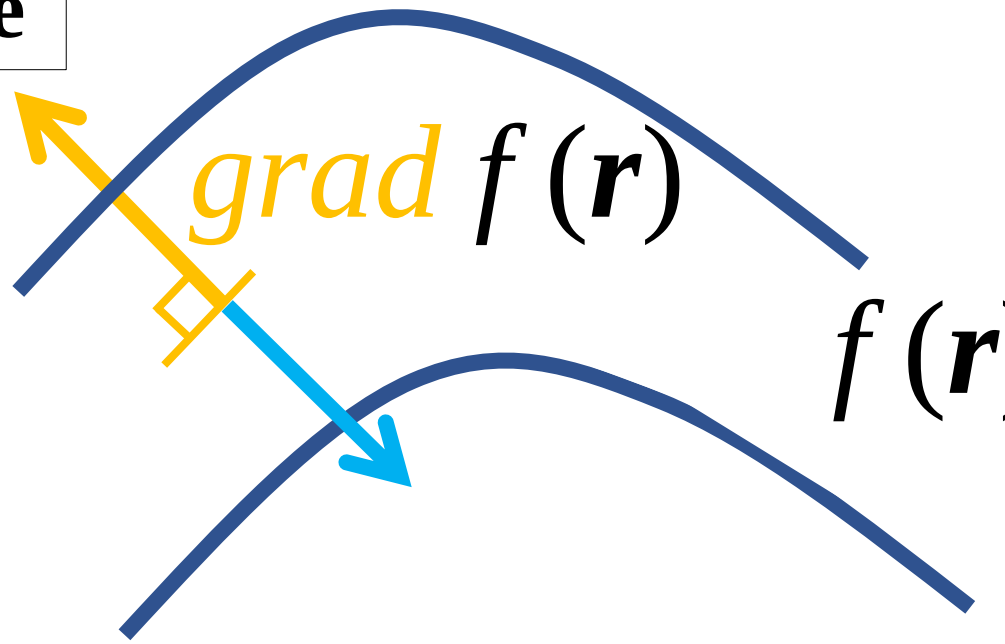
$$= grad\, f(\boldsymbol{r})$$

$$= {}^t\left(\frac{\partial E}{\partial x}, \frac{\partial E}{\partial y}, \frac{\partial E}{\partial z} \right)$$

$$\nabla f(\boldsymbol{r}) = grad\, f(\boldsymbol{r})$$

Gradient (grad $f(r)$)
Steepest Descent Minimization of Multivariable Function

Iso-Surface



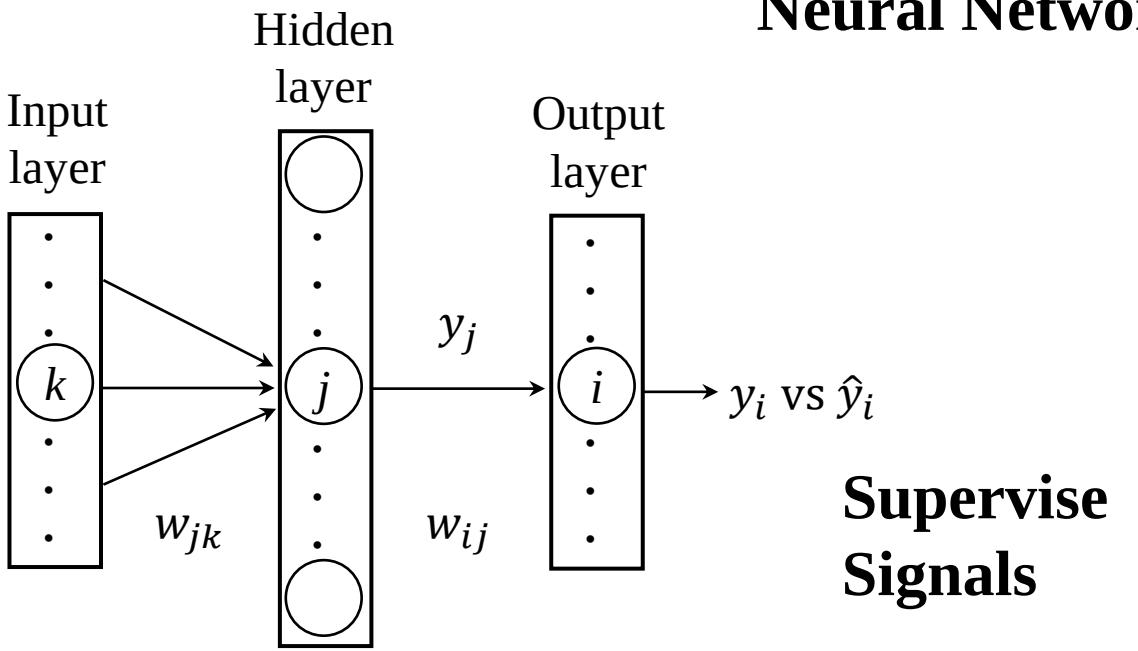
$$f(\boldsymbol{r}) = E_1 \quad \textbf{(constant)}$$

$$\boldsymbol{r} = {}^t(x, y, z)$$

$$\nabla f(\boldsymbol{r}) = grad\, f(\boldsymbol{r}) = {}^t\left(\frac{\partial E}{\partial x}, \frac{\partial E}{\partial y}, \frac{\partial E}{\partial z} \right)$$

Backpropagation

Multi Layer Perceptron-Type Neural Network



Error Function / Loss Function

$$E = \frac{1}{2}\sum_i (y_i - \hat{y}_i)^2 + \boldsymbol{R(w)}$$

Training/Learning（学習）における **Overfitting**（過学習）を回避するために、**Regularization**（正則化）が重要な役割を果たす。
→ L1 Norm, L2 Norm などがよく用いられる。

Overfitting： Training Dataset に対しては正しく出力するが、未知のTest Dataset には誤った出力を与えること。

$$\nabla f(\boldsymbol{r})$$

$$= grad\, f(\boldsymbol{r})$$

$$= {}^{t}\left(\frac{\partial E}{\partial x}, \frac{\partial E}{\partial y}, \frac{\partial E}{\partial z} \right)$$

$$\sum_k w_{jk}x_k = x_j$$
$$y_j = \sigma(x_j)$$
$$x_i = \sum_j w_{ij}y_j$$
$$y_i = \sigma(x_i)$$

Backpropagation

$$\frac{dE}{dt} = \sum_i \frac{\partial E}{\partial w_{ij}}\frac{dw_{ij}}{dt}$$

$$\frac{dw_{ij}}{dt} = -\frac{\partial E}{\partial w_{ij}} \dots (*)$$

Error Function / Loss Function

$$E = \frac{1}{2}\sum_i (y_i - \hat{y}_i)^2$$

Chain Rule により

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial x_i}\frac{\partial x_i}{\partial w_{ij}} = (y_i - \hat{y}_i)\sigma(x_i)\big(1 - \sigma(x_i)\big)y_j$$

$$\left(\because) \quad \frac{\partial x_{ij}}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}}\sum_j w_{ij}y_j \text{ であり，} w_{ij} \text{で偏微分するので，} y_j \text{のみ残る}\right)$$

よって，$(*)$は

$$\frac{dw_{ij}}{dt} = -(y_i - \hat{y}_i)\sigma(x_i)\big(1 - \sigma(x_i)\big)y_j$$
$$\frac{\Delta w_{ij}}{\Delta t} = -(y_i - \hat{y}_i)\sigma(x_i)\big(1 - \sigma(x_i)\big)y_j$$

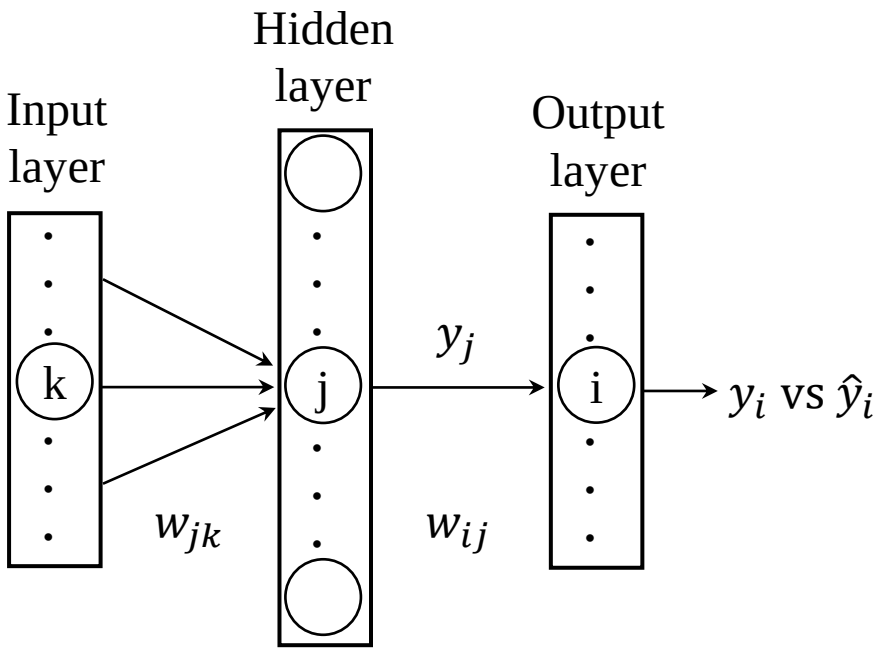
$$\Delta w_{ij} = w_{ij}^{(n+1)} - w_{ij}^{(n)} = -\Delta t(y_i - \hat{y}_i)\sigma(x_i)\big(1 - \sigma(x_i)\big)y_j$$

Backpropagation

$$\sum_k w_{jk}x_k = x_j$$
$$y_j = \sigma(x_j)$$
$$x_i = \sum_j w_{ij}y_j$$
$$y_i = \sigma(x_i)$$

$$\frac{dE}{dt} = \sum_j \frac{\partial E}{\partial w_{jk}} \frac{dw_{jk}}{dt}$$

$$\frac{dw_{jk}}{dt} = -\frac{\partial E}{\partial w_{jk}} \dots (*)$$



Chain Ruleより

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial x_i}\frac{\partial x_i}{\partial y_j}\frac{\partial y_j}{\partial x_j}\frac{\partial x_j}{\partial w_{jk}} = (y_i - \hat{y}_i)\sigma(x_i)\big(1-\sigma(x_i)\big)w_{ij}\sigma(x_j)\Big(1-\sigma\big(x_j\big)\Big)x_k$$

よって，$(*)$は

$$\frac{dw_{jk}}{dt} = -(y_i - \hat{y}_i)\sigma(x_i)\big(1-\sigma(x_i)\big)w_{ij}\sigma(x_j)\Big(1-\sigma\big(x_j\big)\Big)x_k$$
$$\frac{\Delta w_{jk}}{\Delta t} = -(y_i - \hat{y}_i)\sigma(x_i)\big(1-\sigma(x_i)\big)w_{ij}\sigma(x_j)\Big(1-\sigma\big(x_j\big)\Big)x_k$$

$$\Delta w_{ij} = w_{jk}^{(n+1)} - w_{jk}^{(n)} = -\Delta t(y_i - \hat{y}_i)\sigma(x_i)\big(1-\sigma(x_i)\big)w_{ij}\sigma(x_j)\Big(1-\sigma\big(x_j\big)\Big)x_k$$

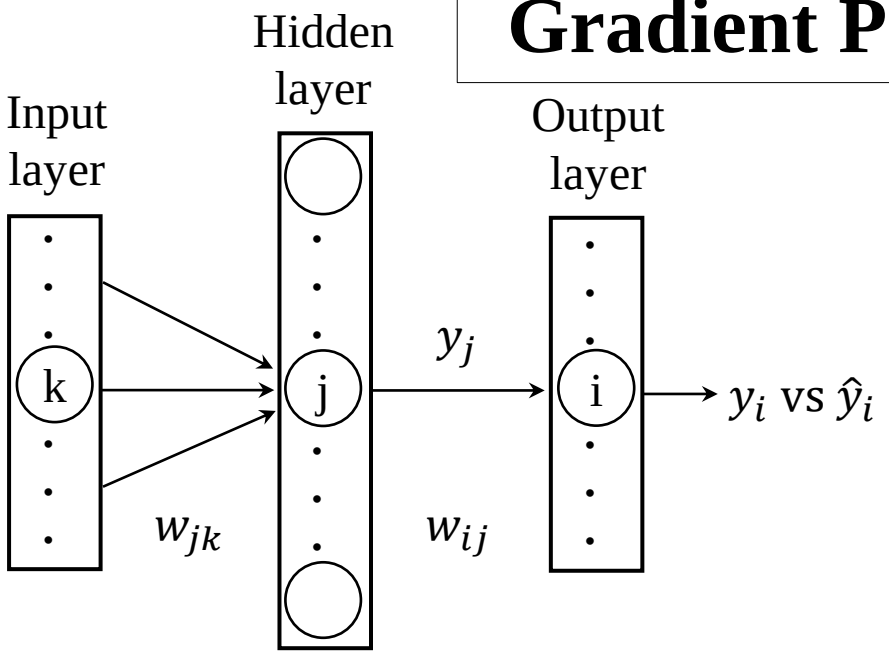
Backpropagation

Vanishing/Exploding Gradient Problems

$$\sum_k w_{jk}x_k = x_j$$
$$y_j = \sigma(x_j)$$
$$x_i = \sum_j w_{ij}y_j$$
$$y_i = \sigma(x_i)$$

$$\frac{dE}{dt} = \sum_j \frac{\partial E}{\partial w_{jk}} \frac{dw_{jk}}{dt}$$

$$\frac{dw_{jk}}{dt} = -\frac{\partial E}{\partial w_{jk}} \dots (*)$$



Chain Ruleより

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial x_i}\frac{\partial x_i}{\partial y_j}\frac{\partial y_j}{\partial x_j}\frac{\partial x_j}{\partial w_{jk}} = (y_i - \hat{y}_i)\sigma(x_i)\big(1-\sigma(x_i)\big)w_{ij}\sigma(x_j)\Big(1-\sigma(x_j)\Big)x_k$$

よって，$(*)$は

$$\frac{dw_{jk}}{dt} = -(y_i - \hat{y}_i)\sigma(x_i)\big(1-\sigma(x_i)\big)w_{ij}\sigma(x_j)\Big(1-\sigma(x_j)\Big)x_k$$
$$\frac{\Delta w_{jk}}{\Delta t} = -(y_i - \hat{y}_i)\sigma(x_i)\big(1-\sigma(x_i)\big)w_{ij}\sigma(x_j)\Big(1-\sigma(x_j)\Big)x_k$$

$$\Delta w_{ij} = w_{jk}^{(n+1)} - w_{jk}^{(n)} = -\Delta t(y_i - \hat{y}_i)\sigma(x_i)\big(1-\sigma(x_i)\big)w_{ij}\sigma(x_j)\Big(1-\sigma(x_j)\Big)x_k$$

Vanishing/Exploding Gradient Problems

勾配の消失と爆発

原因は **Back Propagation** における

Activation Function の微分と Weight の “掛け算”

を繰り返し実行することになるため！　これを回避するためには、以下の３つの点が基本です。

ReLU (used as an Activation Function)

Batch Normalization （Cf. **Mini-Batch**）

Skip Connection (→ Residual Network.　　Cf. ODEnet)

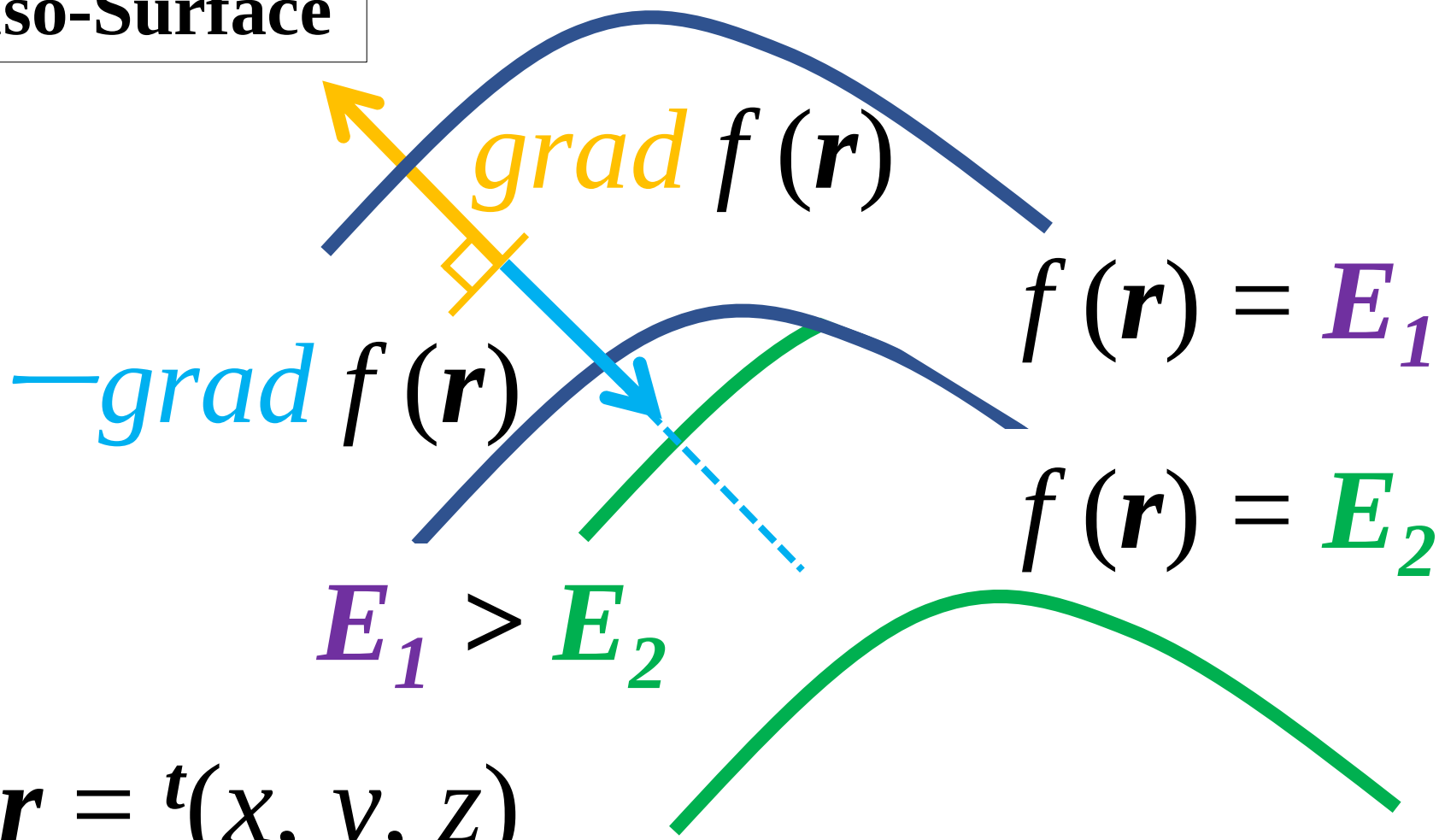
etc.

参考書：　Scikit-Learn、Keras、TensorFlowによる実践機械学習 第2版, Aurélien Géron 著

$\nabla f(\boldsymbol{r})$
$= grad\ f(\boldsymbol{r})$

Gradient (grad $f(r)$)
Steepest Descent Minimization of Multivariable Function

Iso-Surface



Vector, Matrix, and Scalar Product

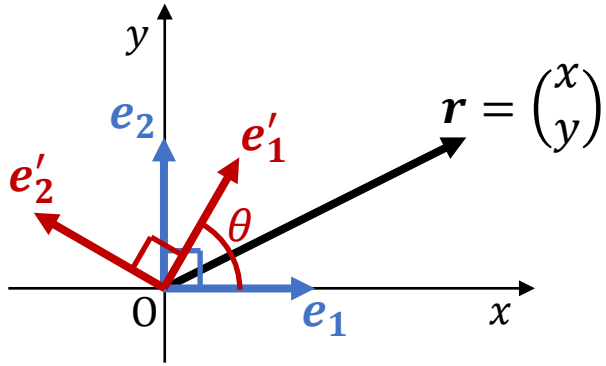
1 Vector

$$\boldsymbol{r} = \begin{pmatrix} x \\ y \end{pmatrix} = x \begin{pmatrix} 1 \\ 0 \end{pmatrix} + y \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$\boldsymbol{e_1}$ $\boldsymbol{e_2}$

"Basis Set (基底)"

$$= \boldsymbol{x}\, \boldsymbol{e_1} + \boldsymbol{y}\, \boldsymbol{e_2}$$



Basis Set: $\{\boldsymbol{e}_i\}_{i=1}^{2}$ を θ だけ回転し，$\{\boldsymbol{e}'_i\}_{i=1}^{2}$ に変化させると，

$$\boldsymbol{r} = x'\boldsymbol{e}'_1 + y'\boldsymbol{e}'_2$$

となり，同じベクトル $\boldsymbol{r}$ であっても，別の座標 x', y' に変更される。

垂直かつ長さ 1 の Basis Set を特に，

"Orthonormal Basis Set" (正規直交基底)

という。より一般に，Basis どうしが直交でないとき，"斜交座標系" という。→ テンソル へ

Vector, Matrix, and Scalar Product

2 Matrix

$$2\left\{\left(\boxed{\boldsymbol{a_1}}\ \ \boxed{\boldsymbol{a_2}}\right) = \boldsymbol{A}\right.$$

$$\underbrace{\qquad\qquad}_{2}$$

$\boldsymbol{A}$: 2 x 2 の matrix

⇒ 2つの成分を持った縦ベクトル $\boldsymbol{a_1}, \boldsymbol{a_2}$ を並べたもの（！）と考える。

前頁の Basis Set $\boldsymbol{e_1}, \boldsymbol{e_2}$ を適用すると,

$$\begin{cases} \boldsymbol{A e_1} = \boldsymbol{A}\begin{pmatrix}1\\0\end{pmatrix} = \boldsymbol{a_1} \\ \boldsymbol{A e_2} = \boldsymbol{A}\begin{pmatrix}0\\1\end{pmatrix} = \boldsymbol{a_2} \end{cases}$$

となり, $\boldsymbol{a_1}, \boldsymbol{a_2}$ を取り出せる。

⇒ Matrix は Vector を並べたもの！

逆に, 上記の式の $\boldsymbol{e_1}, \boldsymbol{e_2}$ を並べて表記すると,

$$\boldsymbol{A}\left(\boxed{\boldsymbol{e_1}}\ \ \boxed{\boldsymbol{e_2}}\right) = \left(\boxed{\boldsymbol{a_1}}\ \ \boxed{\boldsymbol{a_2}}\right)$$

のように, Matrix の形に書き換えられる。

Vector, Matrix, and Scalar Product

より一般に，$\boldsymbol{e_1}$, $\boldsymbol{e_2}$ を任意のベクトル $\boldsymbol{p_1}$, $\boldsymbol{p_2}$ とすると，

$$\begin{cases} \boldsymbol{A}\boldsymbol{p_1} = \boldsymbol{A}\begin{pmatrix} p_1^x \\ p_1^y \end{pmatrix} = \boldsymbol{b_1} \\ \boldsymbol{A}\boldsymbol{p_2} = \boldsymbol{A}\begin{pmatrix} p_2^x \\ p_2^y \end{pmatrix} = \boldsymbol{b_2} \end{cases}$$

Matrix の掛け算としてまとめると ⇒

$$\boldsymbol{A}\begin{pmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b_1} & \boldsymbol{b_2} \end{pmatrix}$$

⇒ Matrix は Vector を並べたものであることが分かる。

Ex) 2 x 2 の Matrix の掛け算への応用 ⇒ 1 x 2 の縦ベクトル2つに分解して考える

$$\begin{pmatrix} \boldsymbol{a_1} & \boldsymbol{a_2} \end{pmatrix}\begin{pmatrix} 2 & -3 \\ 5 & 4 \end{pmatrix} = \begin{pmatrix} 2\boldsymbol{a_1} + 5\boldsymbol{a_2} & -3\boldsymbol{a_1} + 4\boldsymbol{a_2} \end{pmatrix}$$

それぞれ 1 x 2 の縦ベクトル　　それぞれ 1 x 2 の縦ベクトル

Vector, Matrix, and Scalar Product

Ex)　$\boldsymbol{A}$ が 2 x 2 の Matrix であるとする。

$$\boldsymbol{A}\begin{pmatrix}2\\5\end{pmatrix}=\begin{pmatrix}-3\\4\end{pmatrix},\ \boldsymbol{A}\begin{pmatrix}-2\\3\end{pmatrix}=\begin{pmatrix}1\\-1\end{pmatrix}$$

のとき,

$$\boldsymbol{A}\begin{pmatrix}2 & -2\\5 & 3\end{pmatrix}$$

を求めよ。

解答）　与えられた条件により、容易に

$$\boldsymbol{A}\begin{pmatrix}2 & -2\\5 & 3\end{pmatrix}=\begin{pmatrix}-3 & 1\\4 & -1\end{pmatrix}$$

※　$\boldsymbol{A}=\begin{pmatrix}a & b\\c & d\end{pmatrix}$ と置いて計算しなくても解けるが・・・

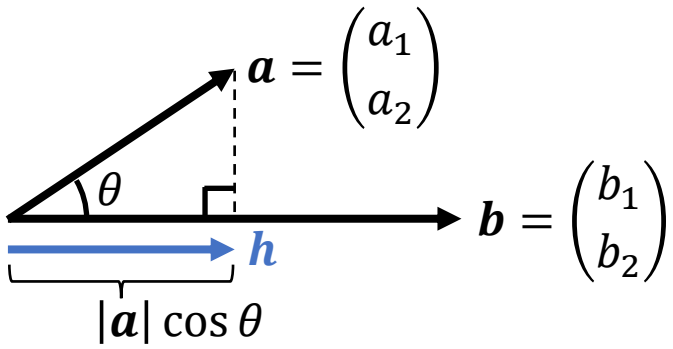
Vector, Matrix, and Scalar Product

3 Scalar Product

$$\boldsymbol{a}\cdot\boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}|\cos\theta$$

⇩ 余弦定理を用いると

$$\boldsymbol{a}\cdot\boldsymbol{b} = a_1b_1 + a_2b_2$$



正射影ベクトル: $\boldsymbol{h} = \dfrac{\boldsymbol{a}\cdot\boldsymbol{b}}{|\boldsymbol{b}|^2}\boldsymbol{b}$

・向きはベクトル $\boldsymbol{b}$ と同じ。

・大きさは，$|\boldsymbol{h}| = \left|\dfrac{\boldsymbol{a}\cdot\boldsymbol{b}}{|\boldsymbol{b}|^2}\boldsymbol{b}\right| = \left|\dfrac{|\boldsymbol{a}||\boldsymbol{b}|\cos\theta}{|\boldsymbol{b}|}\dfrac{\boldsymbol{b}}{|\boldsymbol{b}|}\right| = |\boldsymbol{a}|\cos\theta$ （$\dfrac{\boldsymbol{b}}{|\boldsymbol{b}|} = 1$）

すなわち，ベクトル $\boldsymbol{a}$ と $\boldsymbol{b}$ の内積 $\boldsymbol{a}\cdot\boldsymbol{b}$ は，ベクトル $\boldsymbol{a}$ の正射影ベクトル $\boldsymbol{h}$ とベクトル $\boldsymbol{b}$ により，

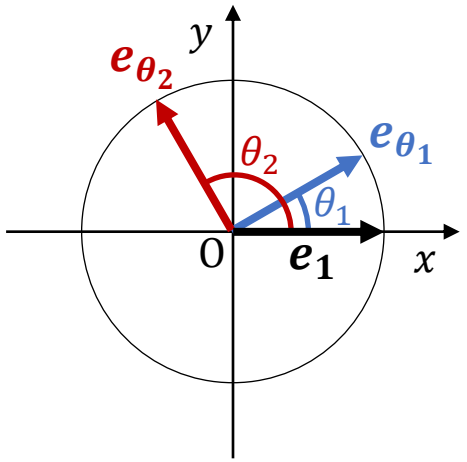
$\boldsymbol{a}\cdot\boldsymbol{b} = |\boldsymbol{h}||\boldsymbol{b}|$ と書くことができる。

Vector, Matrix, and Scalar Product

3 Scalar Product

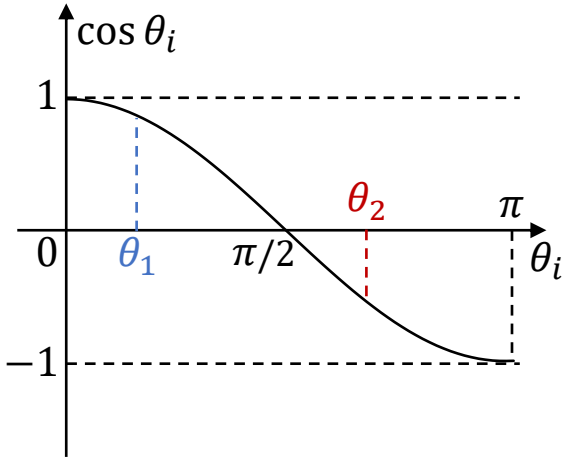
大きさが 1 のベクトル $\boldsymbol{e_1}$ と $\boldsymbol{e_{\theta_i}}$ を考えると,

$$\boldsymbol{e_1} \cdot \boldsymbol{e_{\theta_i}} = \cos\theta_i$$



$\theta_i = 0$ のとき, $\boldsymbol{e_{\theta_i}} = \boldsymbol{e_1}$ となり $\cos\theta_i$ は最大。
$\theta_i = \pi$ のとき, $\boldsymbol{e_{\theta_i}} = -\boldsymbol{e_1}$ となり $\cos\theta_i$ は最小。

⇒ 内積 $\boldsymbol{e_1} \cdot \boldsymbol{e_{\theta_i}}$ は,
ベクトル $\boldsymbol{e_1}$ と $\boldsymbol{e_{\theta_i}}$ との "類似性"



Vector, Matrix, and Scalar Product

$$A = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \qquad B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

$$^tA = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \end{pmatrix} \qquad ^tB = \begin{pmatrix} b_1 & b_2 & \cdots & b_n \end{pmatrix}$$

A と ^tB の積

$$A\ {}^tB = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \begin{pmatrix} b_1 & b_2 & \cdots & b_n \end{pmatrix} = \begin{pmatrix} a_1 \cdot b_1 & \cdots & a_1 \cdot b_n \\ \vdots & \ddots & \vdots \\ a_n \cdot b_1 & \cdots & a_n \cdot b_n \end{pmatrix}$$

B と ^tA の積

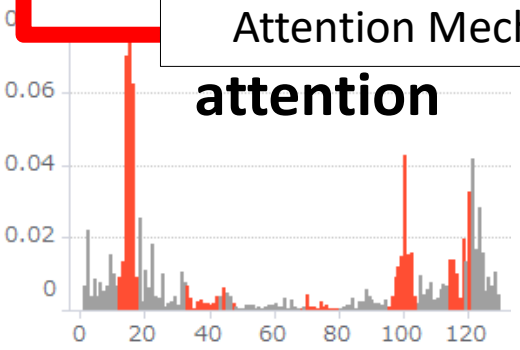
$$B\ {}^tA = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_n \end{pmatrix} = \begin{pmatrix} b_1 \cdot a_1 & \cdots & b_1 \cdot a_n \\ \vdots & \ddots & \vdots \\ b_n \cdot a_1 & \cdots & b_n \cdot a_n \end{pmatrix}$$

Vector, Matrix and Scalar Product

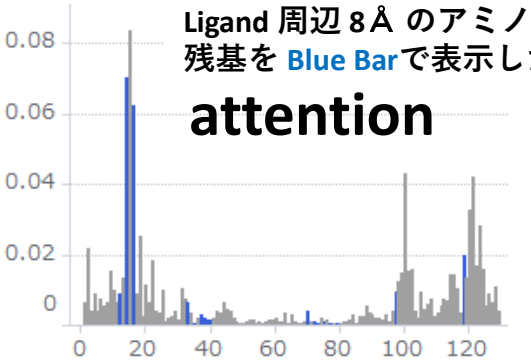
$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

か？

注）これは Multi-Head Attention で、他の Attention Mechanisms と区別しましょう



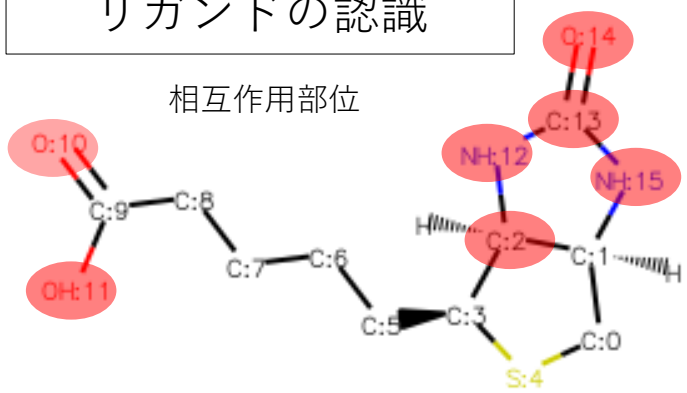
アミノ酸残基

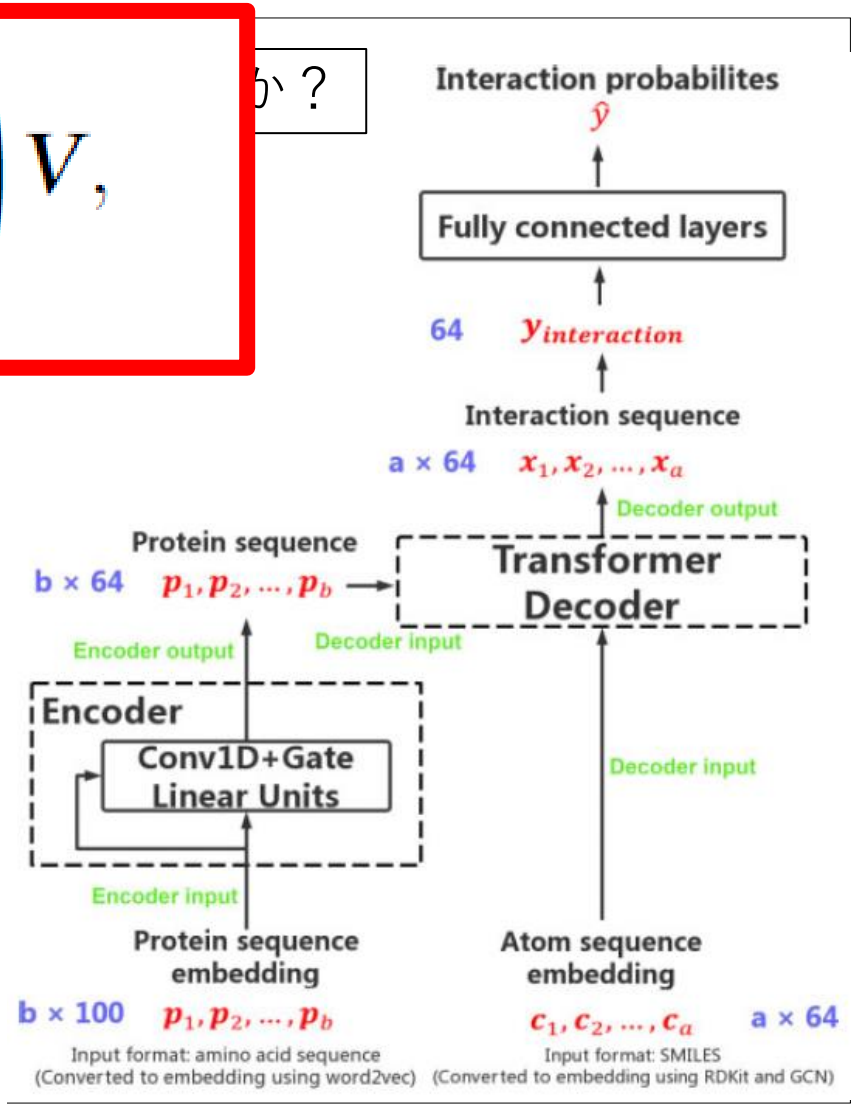


アミノ酸残基

リガンドの認識

相互作用部位





アミノ酸残基をベクトル化

化合物の原子をベクトル化

人工知能（AI）による創薬とは？

創薬のあらゆる領域において、 AI が既に活用されている！

今日の Topics は？

分子設計（I）

タンパク質の**アミノ酸配列**から、結合サイトとリガンドを Identify（Affinity）

分子設計（II）

タンパク質・化合物の複合体の **3D Structural Docking** （& Affinity）

医薬品の**電子構造**

薬理効果、PK and PD、安全性、分子設計（III）、etc.

→ 物質の様々な性質（Properties）や生体反応（Reactivity）を理解するための基盤

→ Screening and Reactivity などの探索における基盤

創薬のためのターゲットの探索　→ Genome Science の応用も！

Transcriptome, **Proteome**, **Interactome**, **Epigenetics**, etc.

→ **Multi Omics** Analysis

系列データ（自然言語処理 [NLP], etc.）

RNN, LSTM,

Transformer

画像データ

CNN

（今日はお話しする余裕がなさそうです・・・）

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

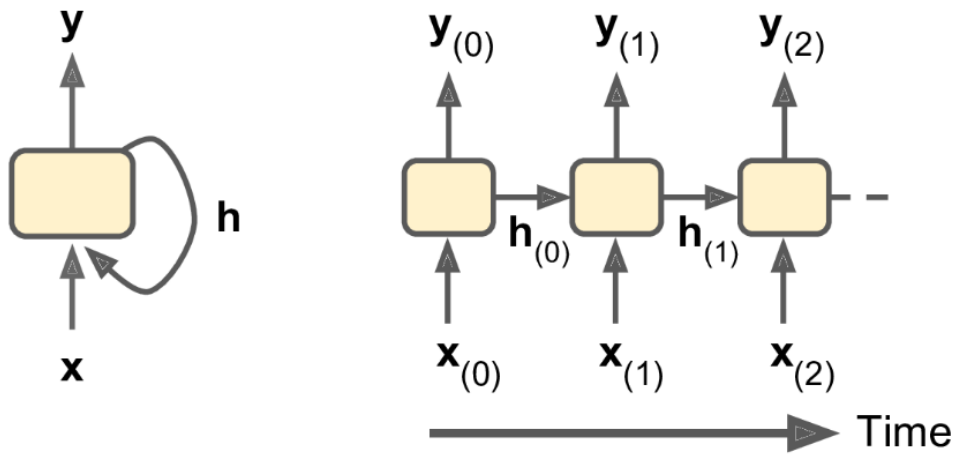
Recurrent Neural Network (RNN)

1. Overviews and Aims of the Techniques

Neural Networks Capable of Handling Sequence Data or Time Series Data.

2. Theory and Network Architecture

Previous Outputs to be Used as Inputs While Having Hidden States.



"Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow." 2nd Edition

Advantages	Drawbacks
• Possibility of processing input of any length • Model size not increasing with size of input • Computation takes into account historical information • Weights are shared across time	• Computation being slow • Difficulty of accessing information from a long time ago • Cannot consider any future input for the current state

Recurrent Neural Network (RNN)

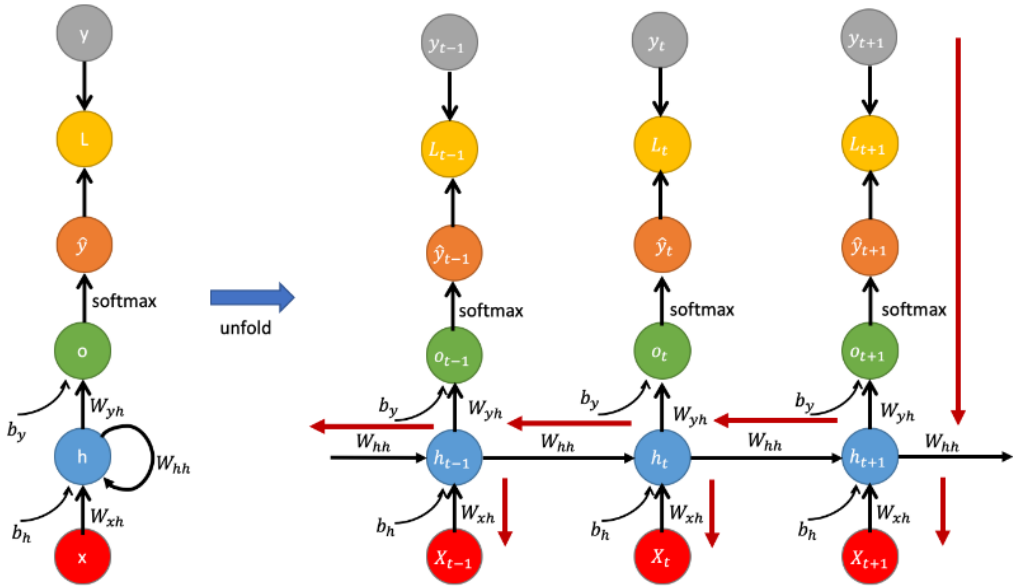
Highlight of the Theory

Forward

$$\boldsymbol{h_t} = \phi_h\left(\boldsymbol{W_{xh}^T X_t} + \boldsymbol{W_{hh}^T h_{t-1}} + \boldsymbol{b_h}\right)$$

$$\hat{y}_t = \phi_o\left(\boldsymbol{W_{yh}^T h_t} + \boldsymbol{b_y}\right)$$

Weight is shared across time.
Activation Function ϕ_h is usual to use a saturating nonlinear function (such as sigmoid or a tanh)



Backward

$$L = \sum_{k=1}^{T} L(y_k, \hat{y}_k),$$

$$\frac{\partial L_k}{\partial \boldsymbol{W}} = \sum_{k=1}^{T} \frac{\partial L_k}{\partial \boldsymbol{W}} = \sum_{k=1}^{T} \left(\frac{\partial L_k}{\partial \boldsymbol{X_t}} \frac{\partial \boldsymbol{X_t}}{\partial \boldsymbol{X_k}} \frac{\partial \boldsymbol{X_k}}{\partial \boldsymbol{W}}\right)$$

$$\frac{\partial \boldsymbol{X_t}}{\partial \boldsymbol{X_k}} = \prod_{i=t}^{k} \frac{\partial \boldsymbol{X_i}}{\partial \boldsymbol{X_{i-1}}} = \prod_{i=t}^{k} \boldsymbol{W_{hh}^T} diag\left(\phi_h'(\boldsymbol{X_{i-1}})\right) \Rightarrow \left\|\phi_h'(\boldsymbol{X_t})\right\| < 1 \text{ (tanh} < 1 \text{ , sigmoid} < 1/4)$$

Vanishing/Exploding Gradient Problems

↓

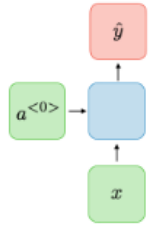
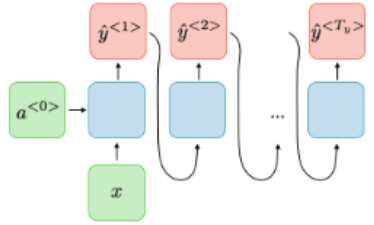
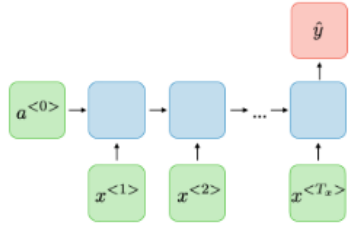
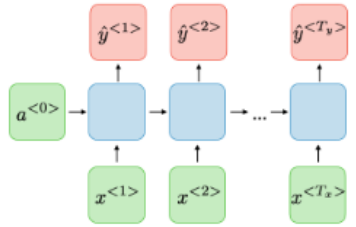
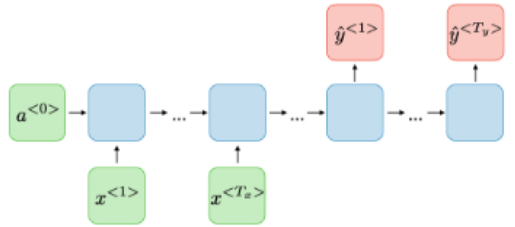
Gradient clipping
Gated RNN (GRU, LSTM etc.)

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Recurrent Neural Network (RNN)

Applications

RNN models are mostly used in the fields of Natural Language Processing (NLP) and speech recognition. Various applications of RNN are summarized in the right table.

Type of RNN	Illustration	Example
One-to-one $T_x = T_y = 1$		Traditional neural network
One-to-many $T_x = 1, T_y > 1$		Music generation
Many-to-one $T_x > 1, T_y = 1$		Sentiment classification
Many-to-many $T_x = T_y$		Name entity recognition
Many-to-many $T_x \neq T_y$		Machine translation

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Recurrent Neural Network (RNN)

Reference

1) “Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow.” 2nd Edition, OREILLY

2) 「ゼロから作るDeepLearning2」 オライリージャパン

3) “On the difficulty of training Recurrent Neural Networks.” arXiv preprint arXiv:1211.5063v2

4) https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

5) https://mmuratarat.github.io/2019-02-07/bptt-of-rnn

LSTM (Long Short-Term Memory)

Hochreiter, S., Schmidhuber, J., Long short-term memory. *Neural Computation* **1997**, 9, 1735-80.

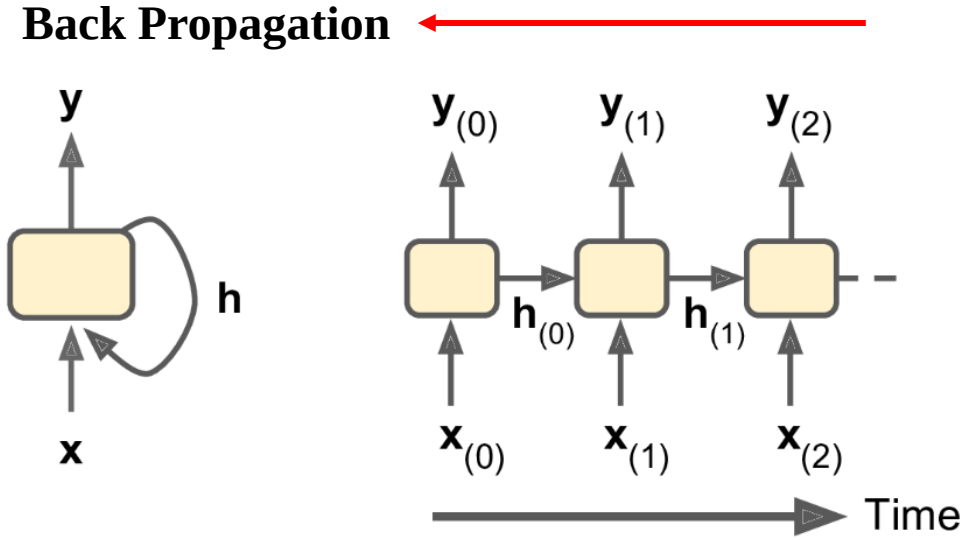
1. Overviews and Aims of the Techniques

- RNN の発展形で、Gated RNN のひとつ

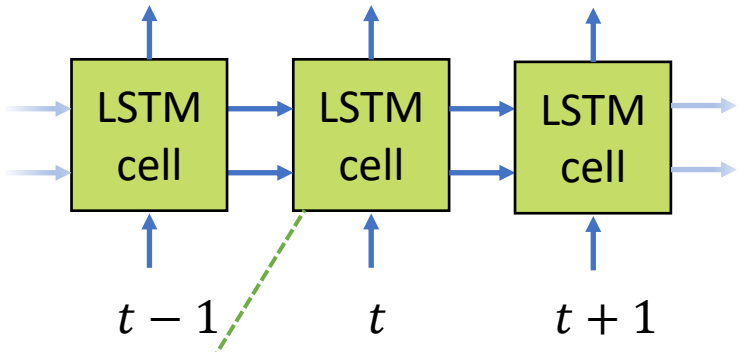
  "The key idea is that the network can learn what to store in the long-term state, what to throw away, and what to read from it." -- Geron, p 516

- Simple RNN の課題である， Back Propagation Through Time での勾配消失に対応

途中で勾配が 0 になると，それ以前の寄与が全て消失する

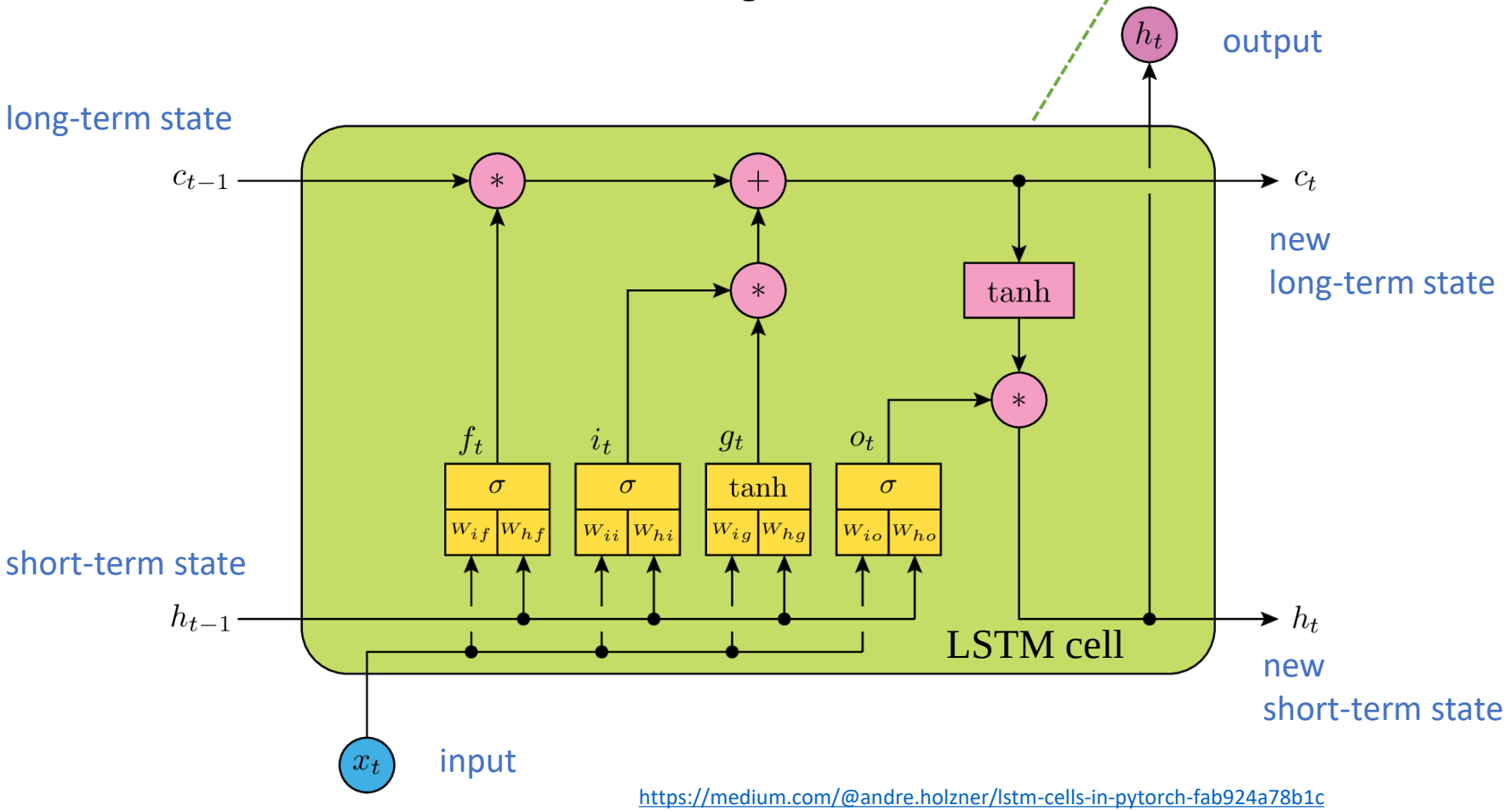


LSTM



2. Theory and Network Architecture

- 3 Gate Controllers & Main Layer
- Gate Controllers も Dense Layer からなり，学習する
- Hidden State Vectors: Short-term State, Long-term State



https://medium.com/@andre.holzner/lstm-cells-in-pytorch-fab924a78b1c

LSTM cell

$$\mathbf{g}_t = \tanh(\mathbf{W}_{ig}{}^{\top}\mathbf{x}_t + \mathbf{b}_{ig} + \mathbf{W}_{hg}{}^{\top}\mathbf{h}_{t-1} + \mathbf{b}_{hg})$$

PyTorch documentation より
https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html

Gates ： 成分 ∈ [0,1]

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ii}{}^{\top}\mathbf{x}_t + \mathbf{b}_{ii} + \mathbf{W}_{hi}{}^{\top}\mathbf{h}_{t-1} + \mathbf{b}_{hi})$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_{if}{}^{\top}\mathbf{x}_t + \mathbf{b}_{if} + \mathbf{W}_{hf}{}^{\top}\mathbf{h}_{t-1} + \mathbf{b}_{hf})$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_{io}{}^{\top}\mathbf{x}_t + \mathbf{b}_{io} + \mathbf{W}_{ho}{}^{\top}\mathbf{h}_{t-1} + \mathbf{b}_{ho})$$

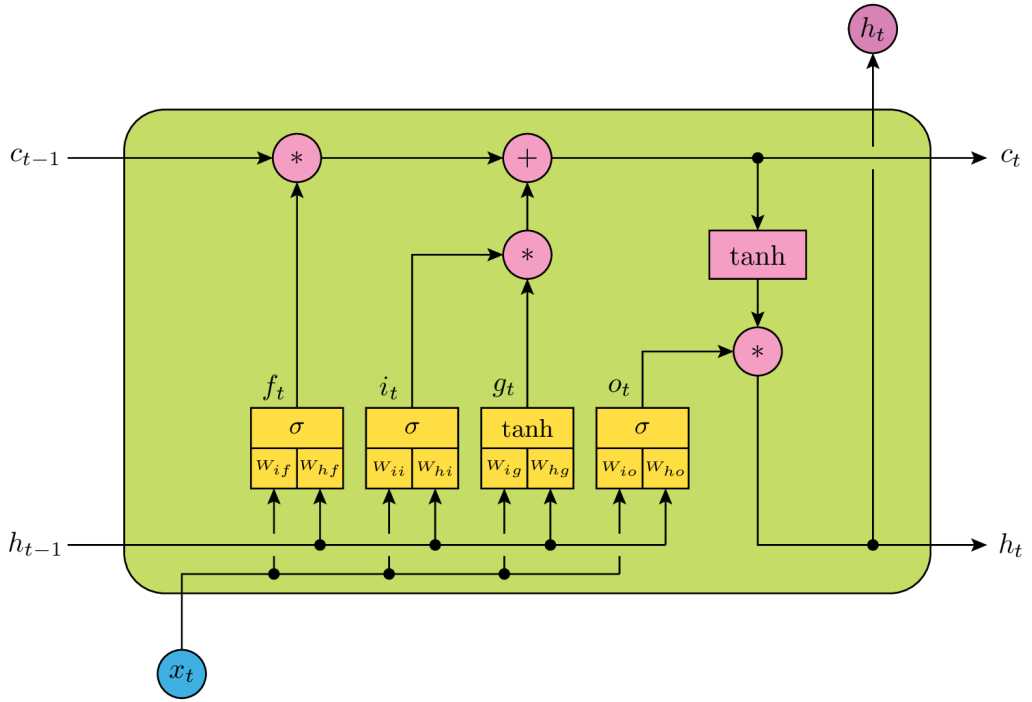
∘: Hadamard product
t: step
σ: sigmoid function

Long-term State

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \mathbf{g}_t$$

Short-term State

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t)$$



---

input 用 weight

$\mathbf{W}_{ig}, \mathbf{W}_{ii}, \mathbf{W}_{if}, \mathbf{W}_{io}$

Short-term State 用 Weight

$\mathbf{W}_{hg}, \mathbf{W}_{hi}, \mathbf{W}_{hf}, \mathbf{W}_{ho}$

Bias

$\mathbf{b}_{ig}$ etc.

Applications of LSTM

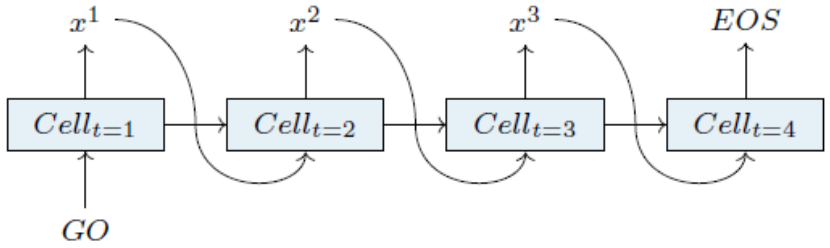
Natural language Processing, speech recognition, handwriting recognition, music composition, market prediction ...

Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, Shahrokh Valaee. Recent Advances in Recurrent Neural Networks. *arXiv preprint*, arXiv:1801.01078, **2017**. https://arxiv.org/abs/1801.01078

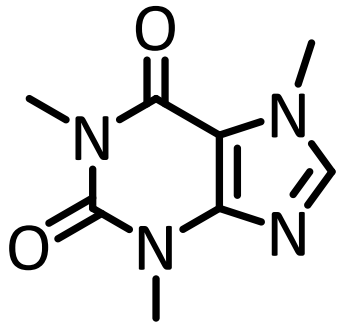
..., and drug design

SMILES
String Molecular Representation

`CN1c2ncn(C)c2C(=O)N(C)C1=O`



Molecular Graph



Reference
REINVENT (AstraZeneca):
Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, Hongming Chen., **Molecular de-novo design through deep reinforcement learning**, *Journal of Cheminformatics* **2017**, 9, 48.

LSTM-Related Architecture

GRU (Gated Recurrent Unit)

- Forget Gate と Input Gate の Controllers が一つにまとまり Update Gate Controller に
- Long-term State と Short-term State Vectors も一つにまとまった
- Update Gate $\mathbf{z}_t$が LSTM の forget gate の役割, $(\mathbf{1}-\mathbf{z}_t)$が input gate の役割
- Reset Gate $\mathbf{r}_t$ は Main Layer に入る Hidden State Vector $\mathbf{h}_{t-1}$ をスケーリングする, そのかわり Output Gate は無い

$$\mathbf{g}_t = \tanh(\mathbf{W}_{ig}{}^{\top}\mathbf{x}_t + \mathbf{b}_{ig} + \mathbf{W}_{hg}{}^{\top}(\mathbf{r}_t \circ \mathbf{h}_{t-1}) + \mathbf{b}_{hg})$$

Reset Gate $\mathbf{r}_t$

$$\mathbf{r}_t = \sigma(\mathbf{W}_{ir}{}^{\top}\mathbf{x}_t + \mathbf{b}_{ir} + \mathbf{W}_{hr}{}^{\top}\mathbf{h}_{t-1} + \mathbf{b}_{hr})$$

Update Gate $\mathbf{z}_t$

$$\mathbf{z}_t = \sigma(\mathbf{W}_{iz}{}^{\top}\mathbf{x}_t + \mathbf{b}_{iz} + \mathbf{W}_{hz}{}^{\top}\mathbf{h}_{t-1} + \mathbf{b}_{hz})$$

output (= new hidden state vector)

$$\mathbf{h}_t = \mathbf{z}_t \circ \mathbf{h}_{t-1} + (\mathbf{1}-\mathbf{z}_t) \circ \mathbf{g}_t$$

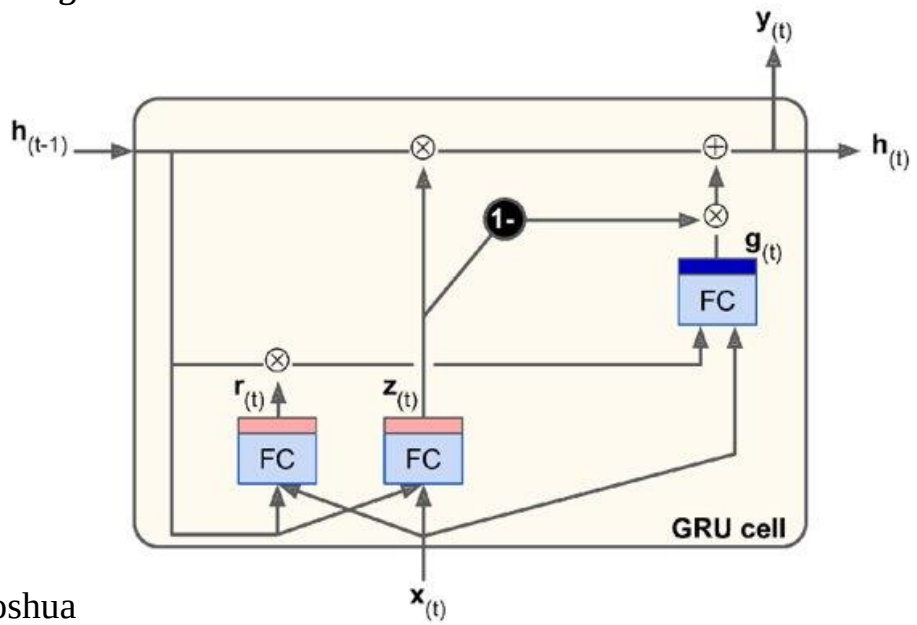
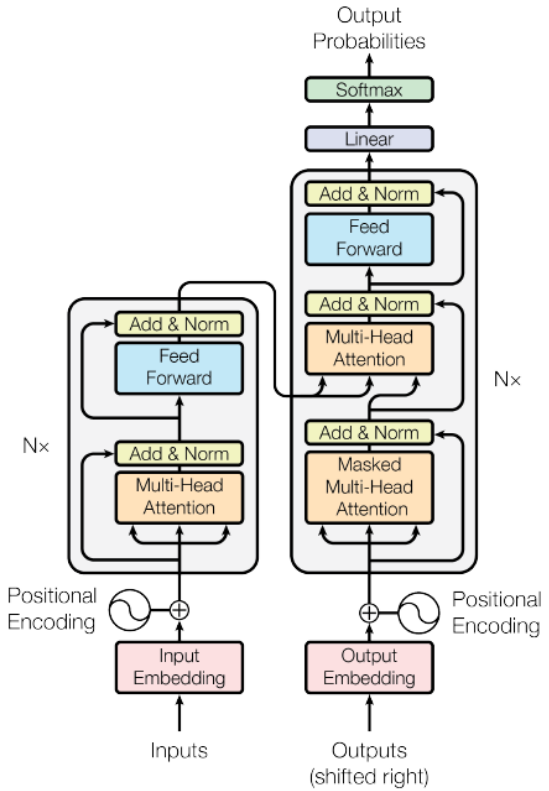


Figure 15-10. GRU cell

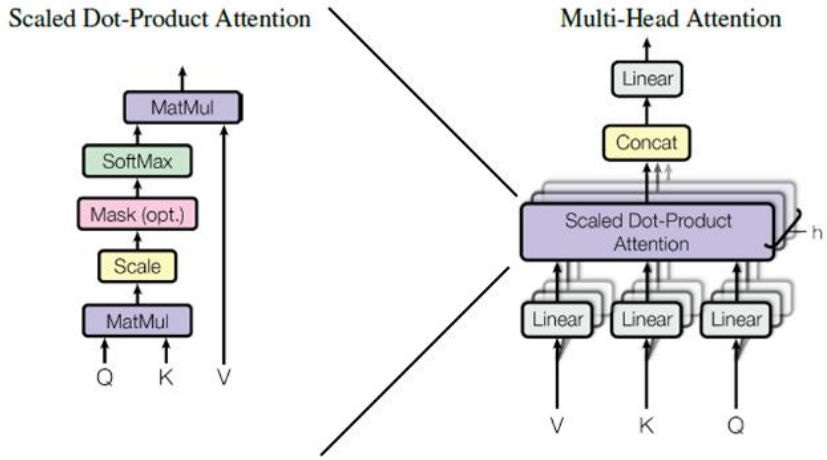
Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. arXiv:1409.1259, **2014**.
https://arxiv.org/abs/1409.1259

Transformer

- **transformer**
  - 自然言語処理（NLP）や画像解析などの広い領域で、大きな成果！
  - **”どこに注意 (attention) するか”** を考慮しながら、AIが学習！



transformer by Google
(**Attention is all you need**)



$$Q_i = QW_i^Q,\ K_i = KW_i^K,\ V_i = VW_i^V$$

$$attention_i = Softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right)V_i$$

$$multi\text{-}head\ attention = concat\left(attention_1, \ldots attention_h\right)W^o$$

Transformer

1. **Overviews and Aims of the Techniques**

   an Encoder-Decoder model involving the Multi-Head Attention without employing RNNs or CNNs.

2. **Theory and Network Architecture**

   The Transformer Architecture is shown below (Figs. 1 and 2).

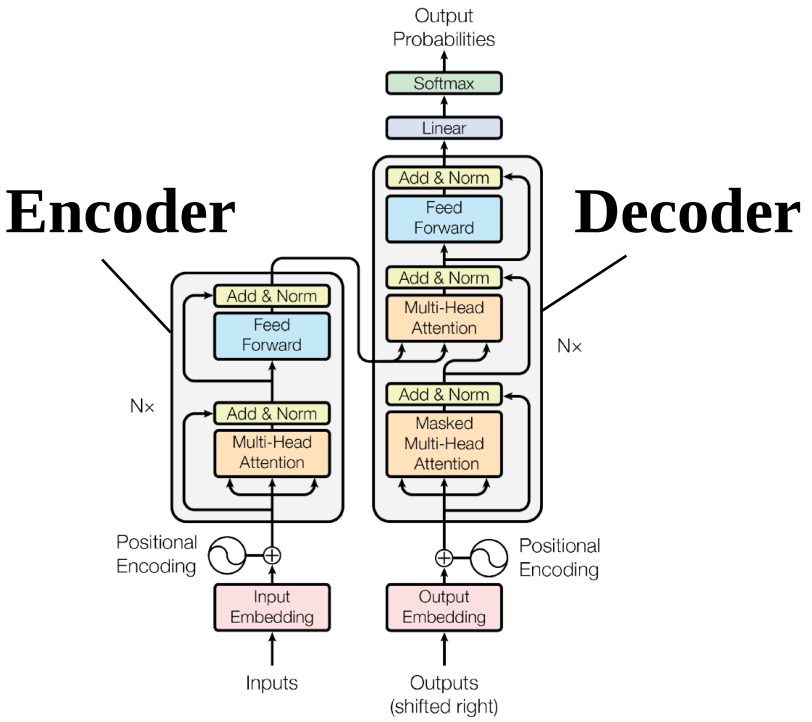
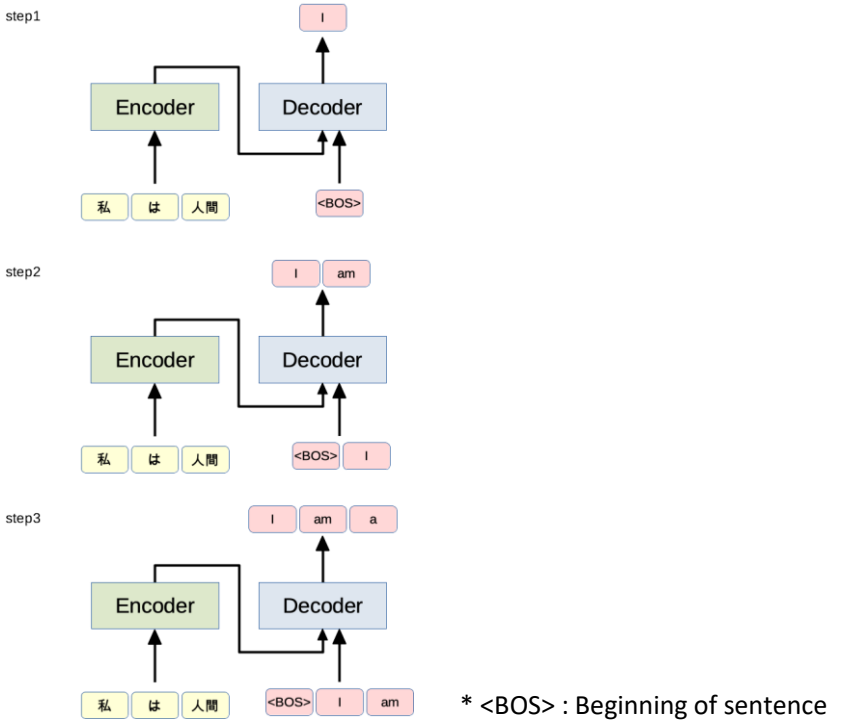


Fig.1.
Architecture of Transformer



* <BOS> : Beginning of sentence

Fig.2
An overview of Translation (Japanese -> English)

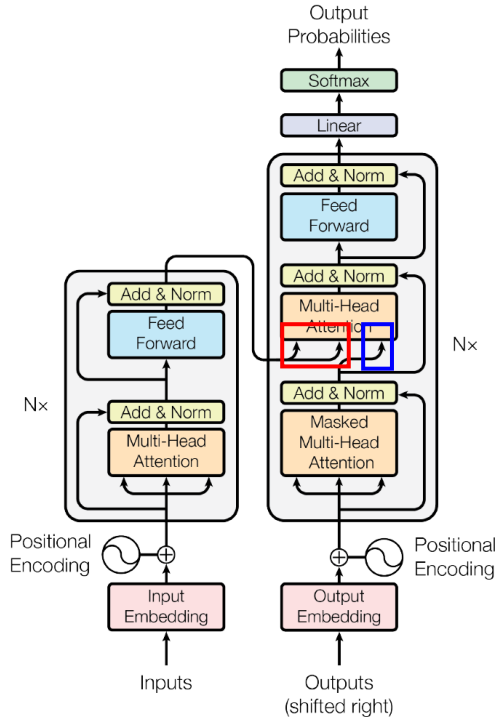
Transformer

Highlight of the Theory-1

Multi-Head Attention

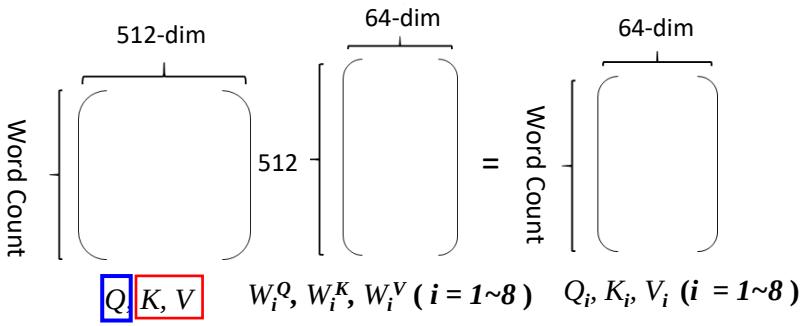
$$\text{Attention}(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Q : Query,
K : Key,
V : Value,
d_k : The dimension of key(512-dim)



Divide Q, K, and V into 8 parts (the number of heads), and calculate the Attention. Finally, they are concatenated in one part.

$$Multihead(Q, K, V) = Concat(head_1, \ldots, headh)W$$
$$\text{where } head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



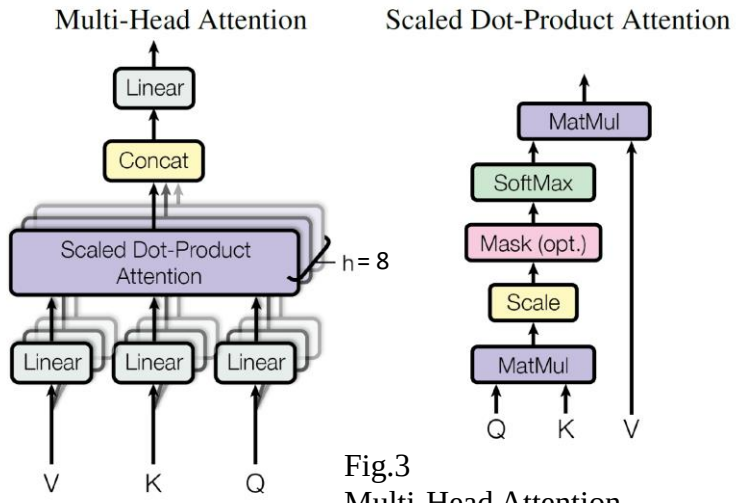


Fig.3
Multi-Head Attention

Transformer

Highlight of the Theory-2

Masked Multi-head Attention

Use(Add) “-inf” to hide future information.
(Prevention of cheating)
Calculation is the same as Multi-head Atten.

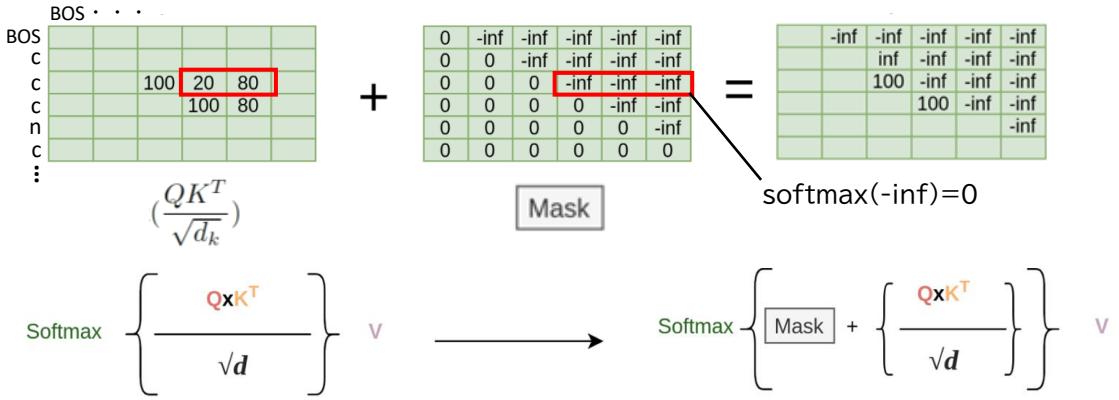


Fig.4
Masked Multi-head Attention

Positional Encoding, Feed-Forward layer

✓(Absolute)Positional Encoding
Add positional information with sinusoid.

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/dmodel})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/dmodel})$$

$\mathbf{d_{model}}$: Number of vector dimensions when word Embedding is performed
i denotes how many dimensions of vector, ***pos*** : the number of vectors

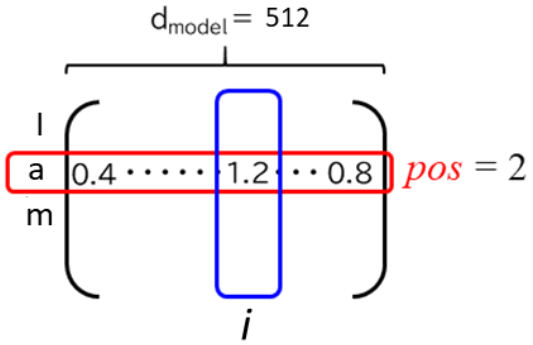
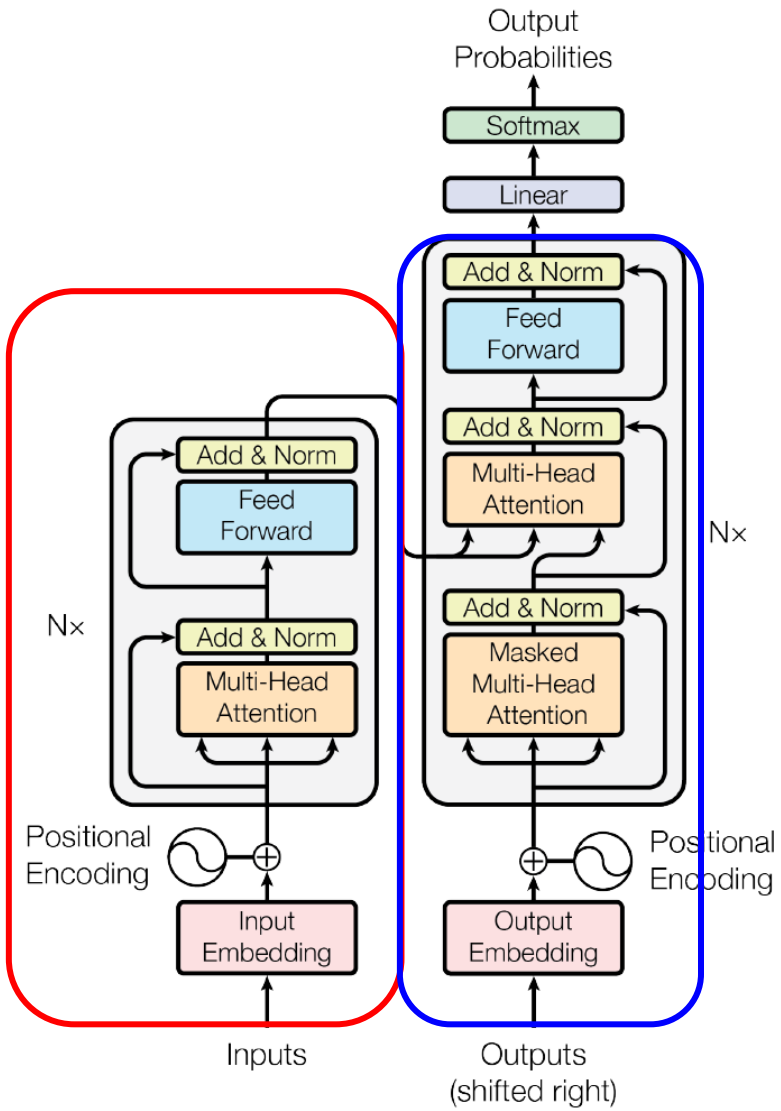


Fig.5
Positional Encoding

✓ Feed-Forward Networks(FFN) layer is a simple linear layer.

$$FFN(x) = \underbrace{\max(0, xW_1 + b_1)}_{\text{ReLU}} W_2 + b_2$$

Molecular Transformer



Encoder側

① Embedding層によって入力化合物を256次元のベクトルに圧縮
② Positional Encoder層によって位置情報を付加
③ Multi-Head Attention層でSelf Attentionを計算し、データ内照応関係を付加
④ 各種Normalizationを行う
⑤ Point-wise順伝播ネットワーク(PFFN)で活性化関数を適用
⑥ 各種Normalizationを行う
③～⑥をN=4層(回)繰り返す

Decoder側

① Embeddingレイヤによって入力化合物を256次元のベクトルに圧縮
② Positional Encoder層によって位置情報を付加
③ Masked Multi-Head AttentionでSelf Attentionを計算し、データ内照応関係を付加
④ 各種Normalizationを行う
⑤ ここまでの出力をQueryに、Encoderの出力をKeyとValueにしてMulti-Head AttentionでAttentionを計算し、異なる時系列データの照応関係情報を獲得
⑥ 各種Normalizationを行う
⑦ PFFNで変換
⑧ 各種Normalizationを行う
③～⑧をN=4層(回)繰り返す

Ref)

1) arXiv:1706.03762v5 [cs.CL] 6 Dec 2017, "Attention Is All You Need"

***2) ACS Cent. Sci.* 2019, 5, 1572−1583 ← Molecular Transformer**

<transformerCPI >

Bioinformatics, 36(16), 2020, 4406–4414
doi: 10.1093/bioinformatics/btaa524
Advance Access Publication Date: 19 May 2020
Original Paper

OXFORD

Structural bioinformatics

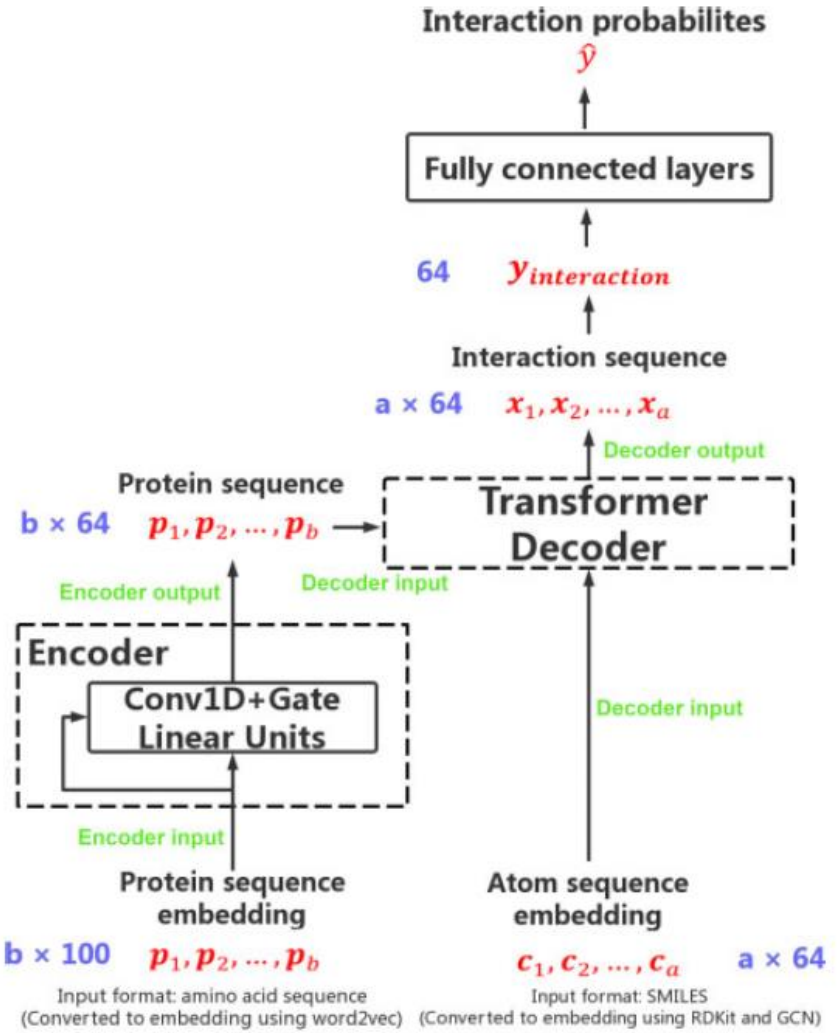
TransformerCPI: improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments

Lifan Chen[1,2], Xiaoqin Tan[1,2], Dingyan Wang[1,2], Feisheng Zhong[1,2], Xiaohong Liu[1,3], Tianbiao Yang[1,2], Xiaomin Luo[1], Kaixian Chen[1,3], Hualiang Jiang[1,3,*] and Mingyue Zheng [1,*]

[1]Drug Discovery and Design Center, State Key Laboratory of Drug Research, Shanghai Institute of Materia Medica, Chinese Academy of Sciences, Shanghai 201203, China, [2]University of Chinese Academy of Sciences, Beijing 100049, China and [3]Shanghai Institute for Advanced Immunochemical Studies, School of Life Science and Technology, ShanghaiTech University, Shanghai 200031, China

Table 1. List of compound atom features

Atom type	C, N, O, F, P, S, Cl, Br, I, other (one hot)
Degree of atom	0, 1, 2, 3, 4, 5, 6 (one hot)
Formal charge	0 or 1
Number of radical electrons	0 or 1
Hybridization type	sp, sp2, sp3, sp3d, sp3d2, other (one hot)
Aromatic	0 or 1
Number of hydrogen atoms attached	0, 1, 2, 3, 4 (one hot)
Chirality	0(False) or 1(True)
Configuration	R, S (one hot)



アミノ酸残基をベクトル化　化合物原子をベクトル化

<transformerCPI >

Bioinformatics, 36(16), 2020, 4406–4414
doi: 10.1093/bioinformatics/btaa524
Advance Access Publication Date: 19 May 2020
Original Paper

Structural bioinformatics

TransformerCPI: improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments

Lifan Chen[1,2], Xiaoqin Tan[1,2], Dingyan Wang[1,2], Feisheng Zhong[1,2], Xiaohong Liu[1,3], Tianbiao Yang[1,2], Xiaomin Luo[1], Kaixian Chen[1,3], Hualiang Jiang[1,3,*] and Mingyue Zheng [1,*]

[1]Drug Discovery and Design Center, State Key Laboratory of Drug Research, Shanghai Institute of Materia Medica, Chinese Academy of Sciences, Shanghai 201203, China, [2]University of Chinese Academy of Sciences, Beijing 100049, China and [3]Shanghai Institute for Advanced Immunochemical Studies, School of Life Science and Technology, ShanghaiTech University, Shanghai 200031, China

Problems

1. **Using inappropriate datasets**
2. **Hidden Ligand Bias**
3. **Splitting dataset inappropriately**

Chen L, Cruz A, Ramsey S, Dickson CJ, Duca JS, Hornak V, et al., "**Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening**", *PLoS ONE*, **14** (2019), e0220113. https://doi.org/10.1371/journal.pone.0220113

RESEARCH ARTICLE

Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening

Lieyang Chen[1,2], Anthony Cruz[1,3], Steven Ramsey[1,2], Callum J. Dickson[4], Jose S. Duca[4], Viktor Hornak[4], David R. Koes[5], Tom Kurtzman[1,2,3]*

1 Department of Chemistry, Lehman College, Bronx, New York, United States of America, **2** Ph.D. program in Biochemistry, The Graduate Center of the City University of New York, New York, United States of America, **3** Ph.D. program in Chemistry, The Graduate Center of the City University of New York, New York, United States of America, **4** Computer-Aided Drug Discovery, Global Discovery Chemistry, Novartis Institutes for Biomedical Research, Cambridge, Massachusetts, United States of America, **5** Department of Computational and System Biology, University of Pittsburgh, Pittsburgh, Pennsylvania, United States of America

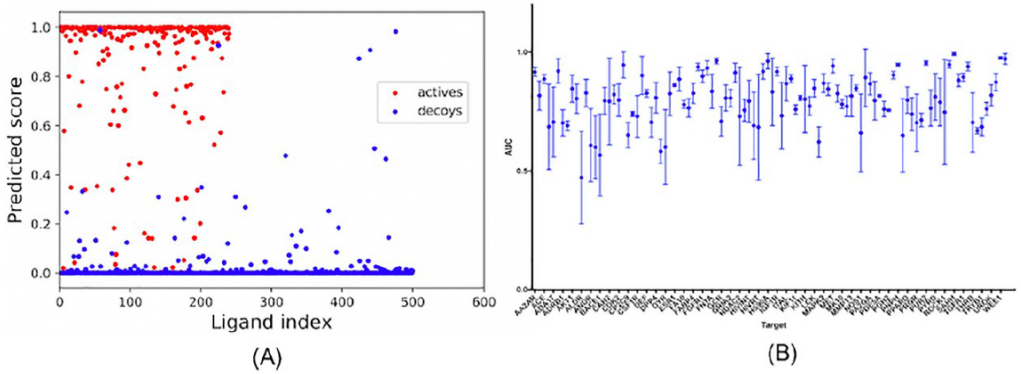


Fig 8. Actives and decoys are generally distinguishable for DUD-E targets. (A) The prediction score of actives and decoys in AA2AR as a representative example; (B) Performance of ligand-trained CNN models trained on small sets of five actives and five decoys. The dots represent mean values, and the bars represent standard deviation.

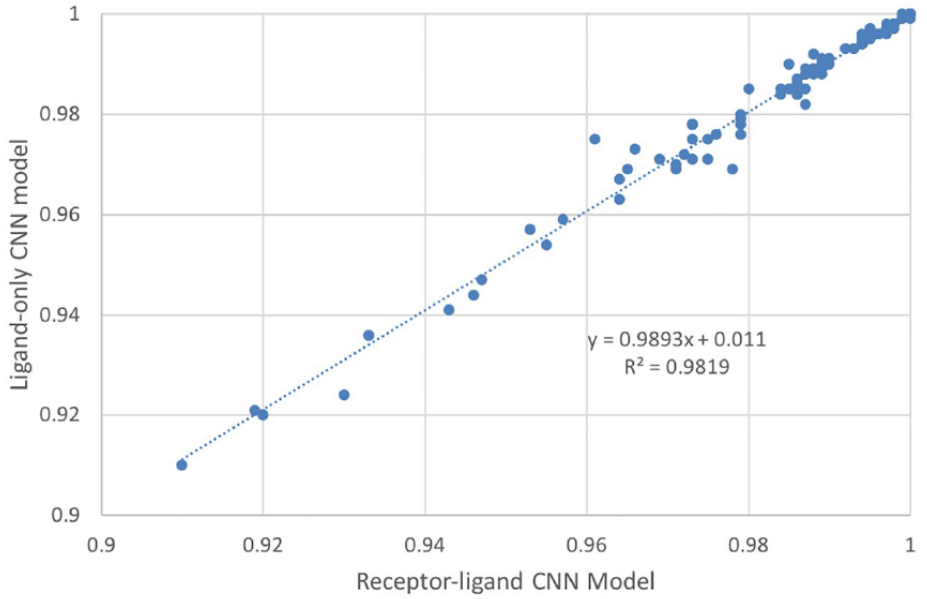


Fig 4. Correlation between the performance of the receptor-ligand CNN model and ligand-only CNN model. The receptor-ligand CNN model was trained on receptor-ligand 3D binding poses, and the ligand-only CNN model was trained on ligand binding poses alone. Each blue dot is a target from DUD-E; there are 102 targets in total.

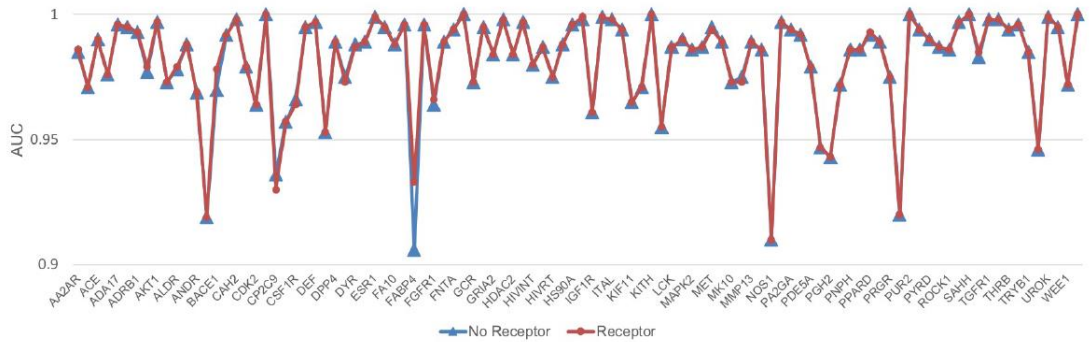


Fig 5. Performance of the receptor-ligand model for the same ligand test sets with and without receptor information. For each target, red dots indicate performance when the receptor structure was provided in the test set, while blue triangles indicate performance when the receptor structure was replaced by a single dummy atom. The x-axis displays each DUD-E target in the same order as they appear in the DUD-E database (http://dude.docking.org/targets). The targets with even indices are not labeled on the x-axis due to space limitations.

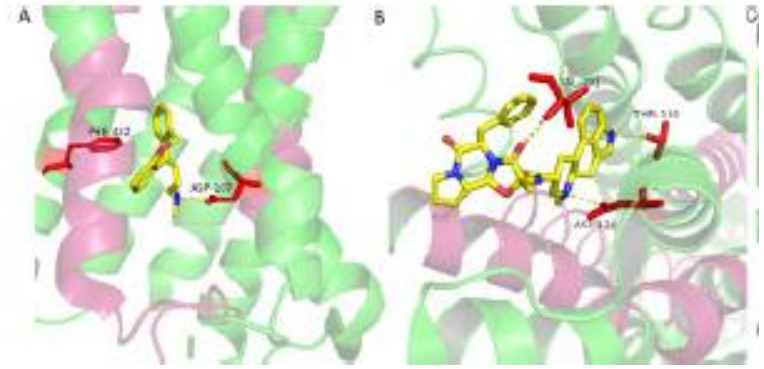
Chen L, Cruz A, Ramsey S, Dickson CJ, Duca JS, Hornak V, et al., "**Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening**", *PLoS ONE*, **14** (2019), e0220113. https://doi.org/10.1371/journal.pone.0220113

<transformerCPI >

Problems

1. Using inappropriate datasets
2. Hidden Ligand Bias
3. Splitting dataset inappropriately

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$



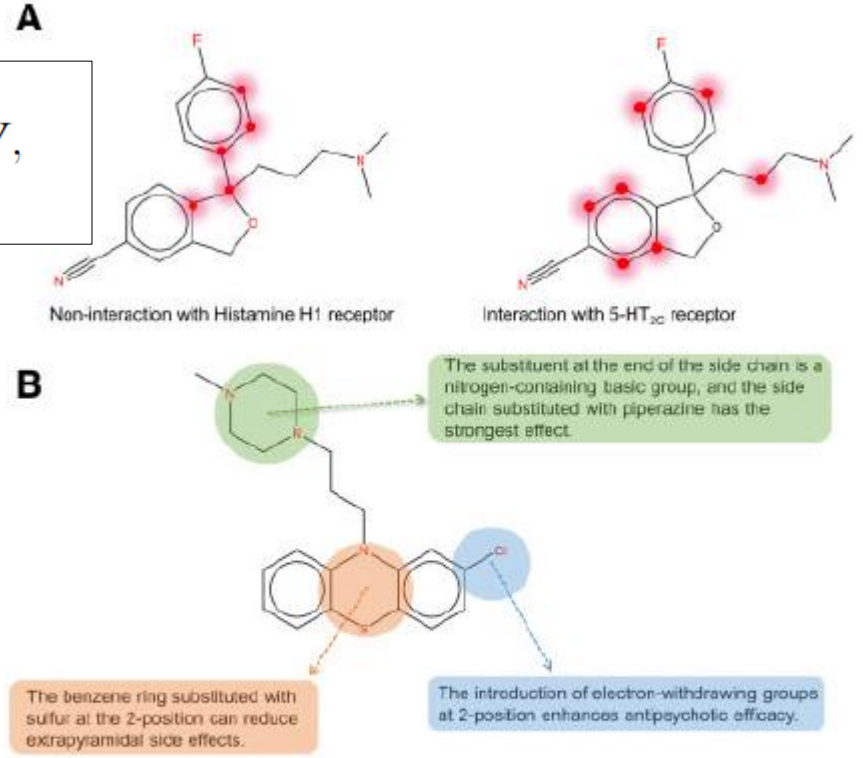


Fig. 4. Attention weights of atoms in different compounds. The atoms, which have high attention scores extracted from TransformerCPI, are highlighted in red. (**A**) A certain ligand shows different attention score distributions when interacting with Histamine H1 receptor and 5-HT_{2C} receptor, respectively. (**B**) The SAR of phenothiazine and its attention scores

Table 4. Comparison results of the proposed model and baselines on human dataset

Method	AUC	Precision	Recall
KNN	0.860	0.927	0.798
RF	0.940	0.897	0.861
L2	0.911	0.913	0.867
SVM	0.910	0.966	0.969
GraphDTA	0.960 ± 0.005	0.882 ± 0.040	0.912 ± 0.040
GCN	0.956 ± 0.004	0.862 ± 0.006	0.928 ± 0.010
CPI–GNN	0.970	0.918	0.923
DrugVQA (VQA-seq)[a]	0.964 ± 0.005	0.897 ± 0.004	0.948 ± 0.003
TransformerCPI	0.973 ± 0.002	0.916 ± 0.006	0.925 ± 0.006

[a]It should be noted that DrugVQA uses protein structural information as input, while its alternative version VQA-seq only using protein sequence information is listed here for a fair comparison.

Table 5. Comparison results of the proposed model and baselines on *C.elegans* dataset

Method	AUC	Precision	Recall
KNN	0.858	0.801	0.827
RF	0.902	0.821	0.844
L2	0.892	0.890	0.877
SVM	0.894	0.785	0.818
GraphDTA	0.974 ± 0.004	0.927 ± 0.015	0.912 ± 0.023
GCN	0.975 ± 0.004	0.921 ± 0.008	0.927 ± 0.006
CPI–GNN	0.978	0.938	0.929
TransformerCPI	0.988 ± 0.002	0.952 ± 0.006	0.953 ± 0.005

Table 6. Comparison results of the proposed model and baselines on BindingDB dataset

Method	AUC	PRC
GraphDTA	0.929	0.917
GCN	0.927	0.913
CPI–GNN	0.603	0.543
TransformerCPI	0.951	0.949

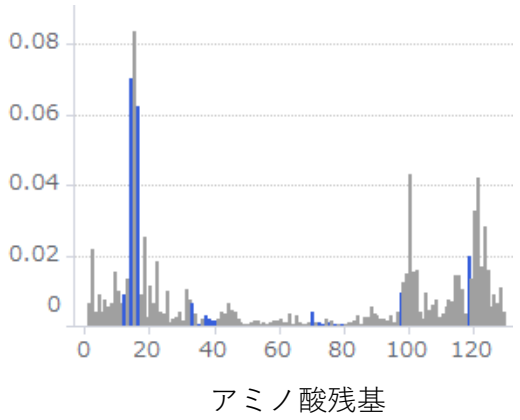
<PL transformer>

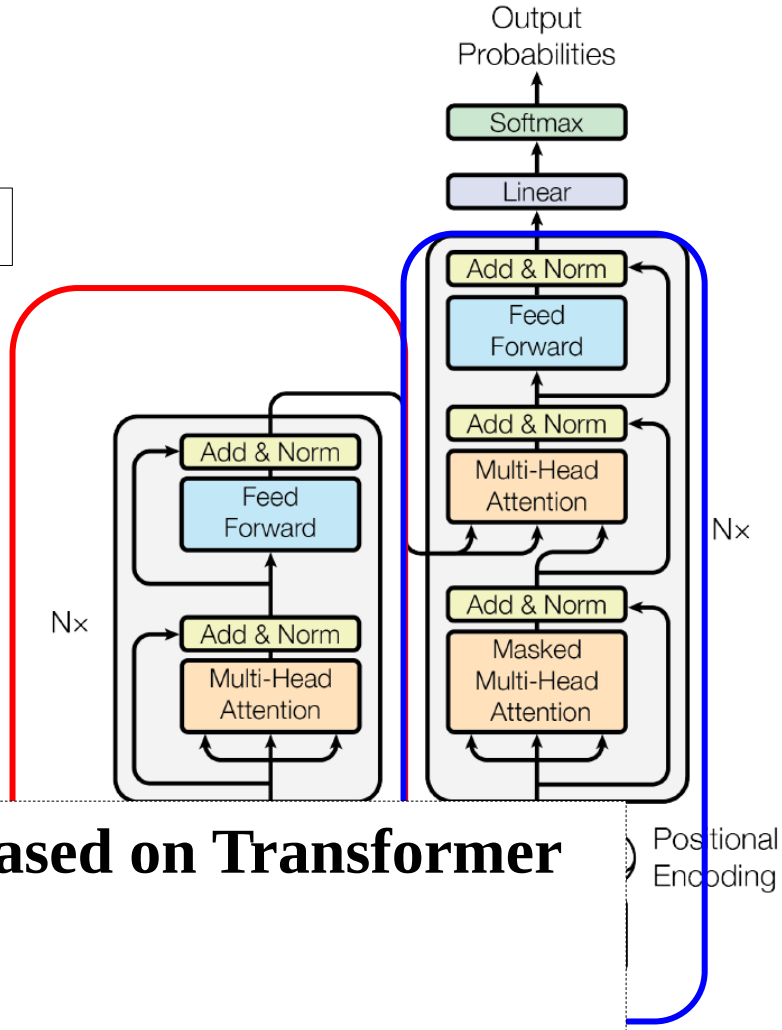
Problems

1. **Using inappropriate datasets**
2. **Hidden Ligand Bias**
3. **Splitting dataset inappropriately**

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

アミノ酸残基の認識

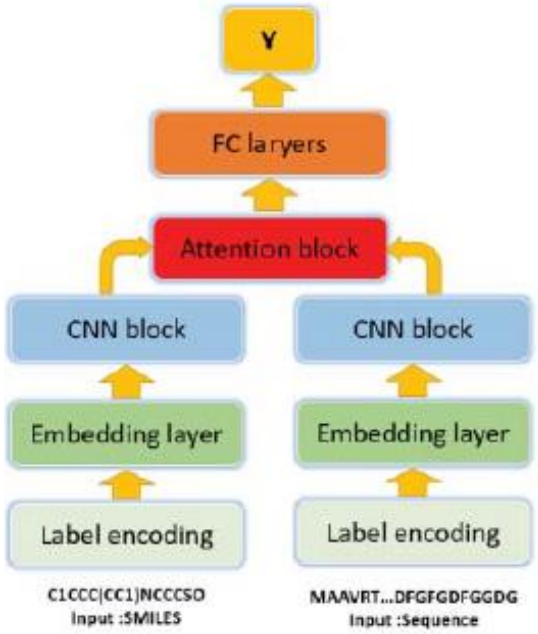




Possible Issues to Construct a New System for P-L Analysis Based on Transformer

1. Involvements of Protein 3D Structures
   → Recognition of Atoms in Side Chains
2. Equivalent Double Recognition of Protein and Ligand
   → Amino Acid Sequences and Compound Chemical Structures
3. Construction of Database of Decoy Ligands
   → Reduction of False Positives and the Bias Derived from Ligands

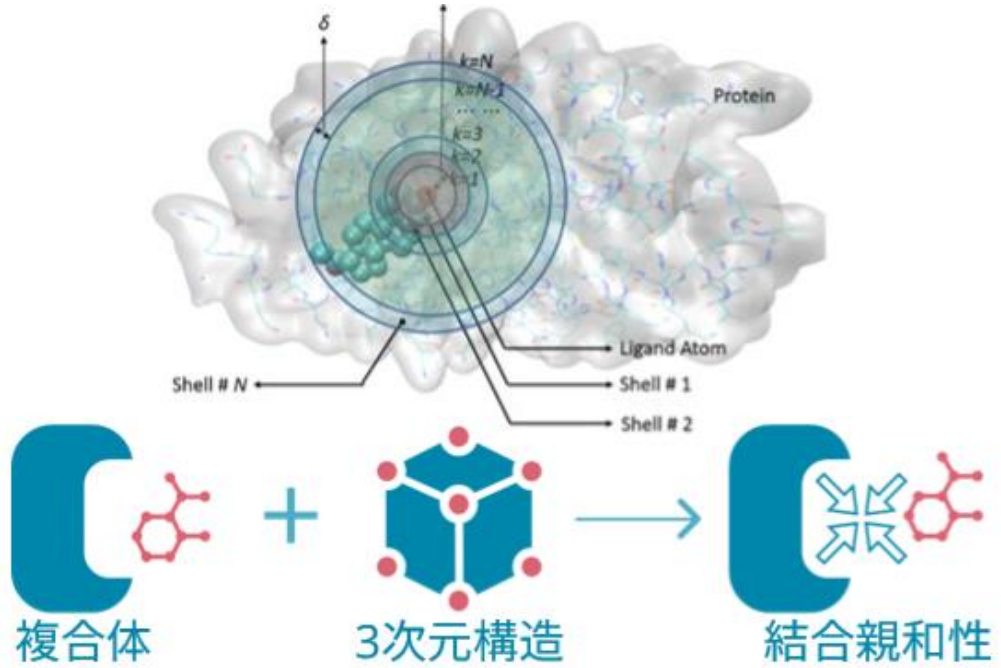
＜他の解析技術＞
transformerCPI 回帰モデルとの比較



AttentionDTA
蛋白質アミノ酸配列、化合物SMILESからWord2vecで特徴抽出して機械学習モデルを構築（Davis datasetのSOTA）

https://github.com/zhaoqichang/AttentionDTA_BIBM

＊Davis dataset : (442kinase) x (68ligand)の活性値30,056データセットで欠損値無し



OnionNet
複合体の 3 次元立体構造を学習データとし、空間的な Shell ごとに、化合物と蛋白質の原子の組み合わせを抽出し、機械学習モデルを構築する

https://github.com/zhenglz/onionnet

参考文献

1) Scikit-Learn、Keras、TensorFlowによる実践機械学習 第2版,

Aurélien Géron 著 （英語版をお勧めします）

2) パターン認識と機械学習, C.M. ビショップ 著 （英語版をお勧めします）

3) ベイズ深層学習 (機械学習プロフェッショナルシリーズ), 須山敦志 著

← **Bayes Statistics** （**AI に確率分布を導入する！**）

Ex) VAE, GPLVM, etc.

4) 曲率とトポロジ, 河野俊丈 著

← **Differential Geometry, Riemann Geometry**

（**AI に “曲がった空間” を導入するために、その基礎を学ぼう！**）

Variational Autoencoder (VAE)

Application

Loss function

$$-\mathcal{L}(\phi,\theta,\boldsymbol{x}) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log q_\phi(\boldsymbol{z}|\boldsymbol{x}) - \log p_\theta(\boldsymbol{z}) - \log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right]$$

$$= D_{KL}\left[q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\boldsymbol{z})\right] - \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]$$

Regularization

Since $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ and $p_\theta(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$,

KL-divergence by represent in closed-form

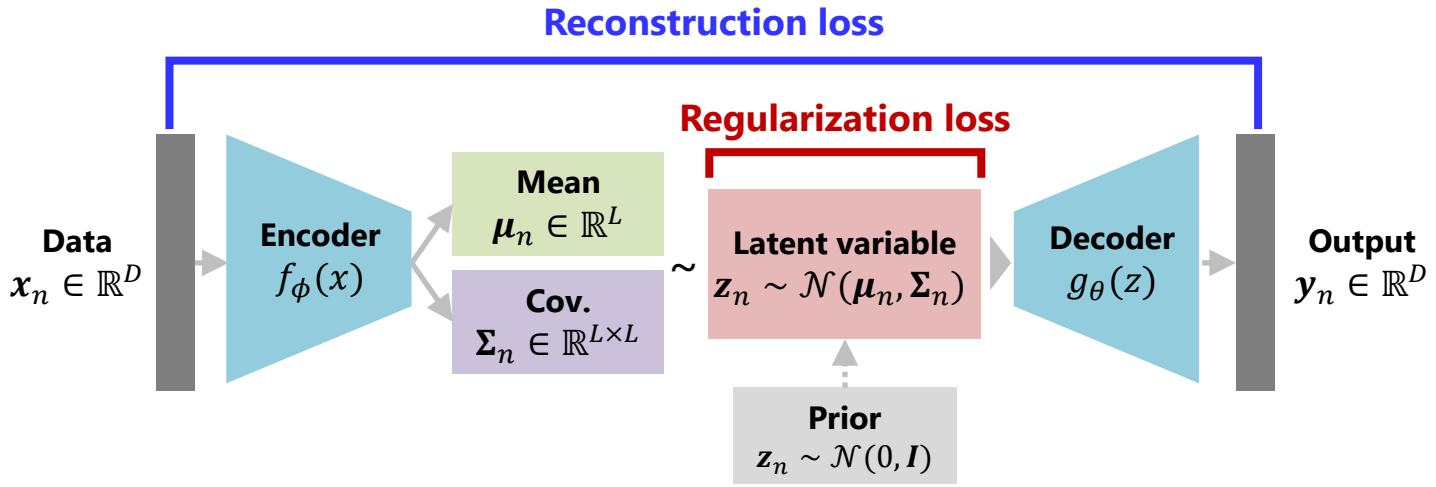
$$D_{KL}\left[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z})\right] = \frac{1}{2}\sum_{n=1}^{N}(tr\boldsymbol{\Sigma} + \boldsymbol{\mu}^\top\boldsymbol{\mu} - D - \log\det\boldsymbol{\Sigma})$$

Reconstruction

Monte carlo estimation

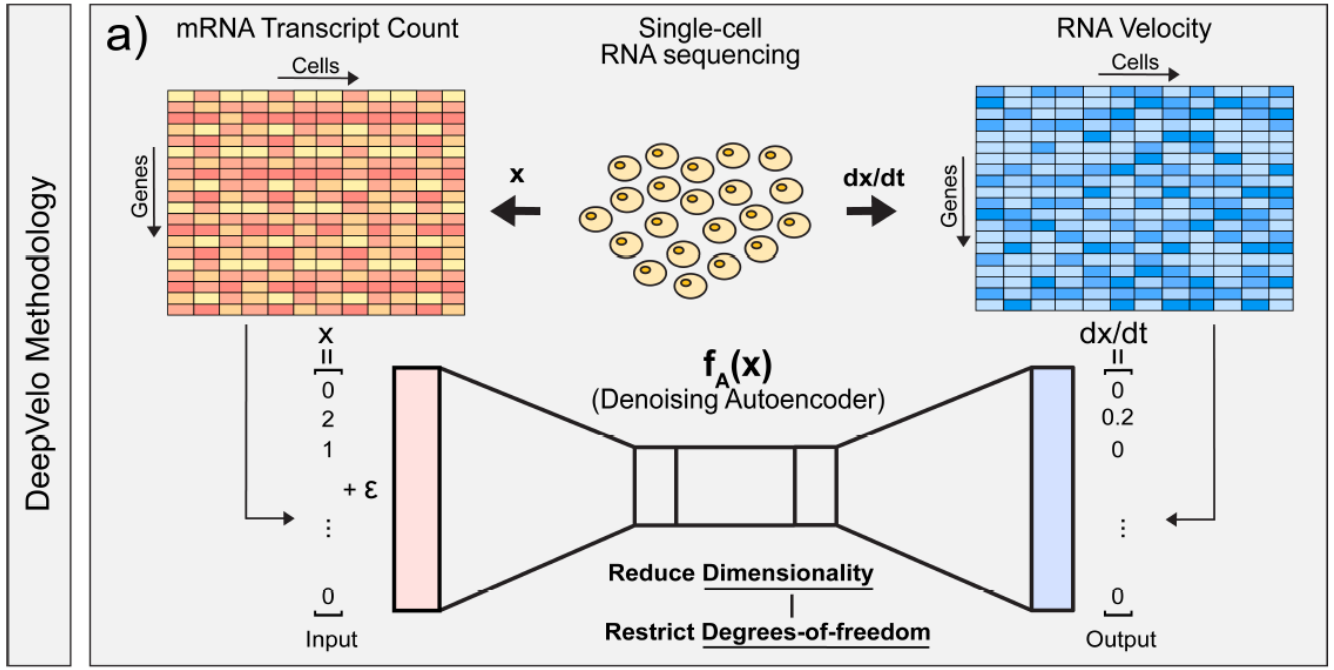
by $\boldsymbol{z}_n \sim q_\phi(\boldsymbol{z}_n|\boldsymbol{x}_n)$

$$\frac{1}{N}\sum_{n=1}^{N} Loss(\boldsymbol{x}_n,\boldsymbol{y}_n)$$



ODEnet I : Genome Science

Application to transcriptomics with VAE-joined ODE model



A Variational Autoencoder (VAE), trained to predict the rate of gene expression change (e.g., RNA velocity is an example of a model representing the turn over of mRNA), is embedded in the ODE, to infer continuous temporal changes in the gene expression within individual cells. The system learns the vector field of gene expression values, and thereby can accurately reproduce the rates.

Zhanlin Chen, William, C. King, Aheyon Hwang, Mark Gerstein, Jing Zhang, "**DeepVelo: Single-cell Transcriptomic Deep Velocity Field Learning with Neural Ordinary Differential Equations**", arXiv, 2022 https://doi.org/10.1101/2022.02.15.480564

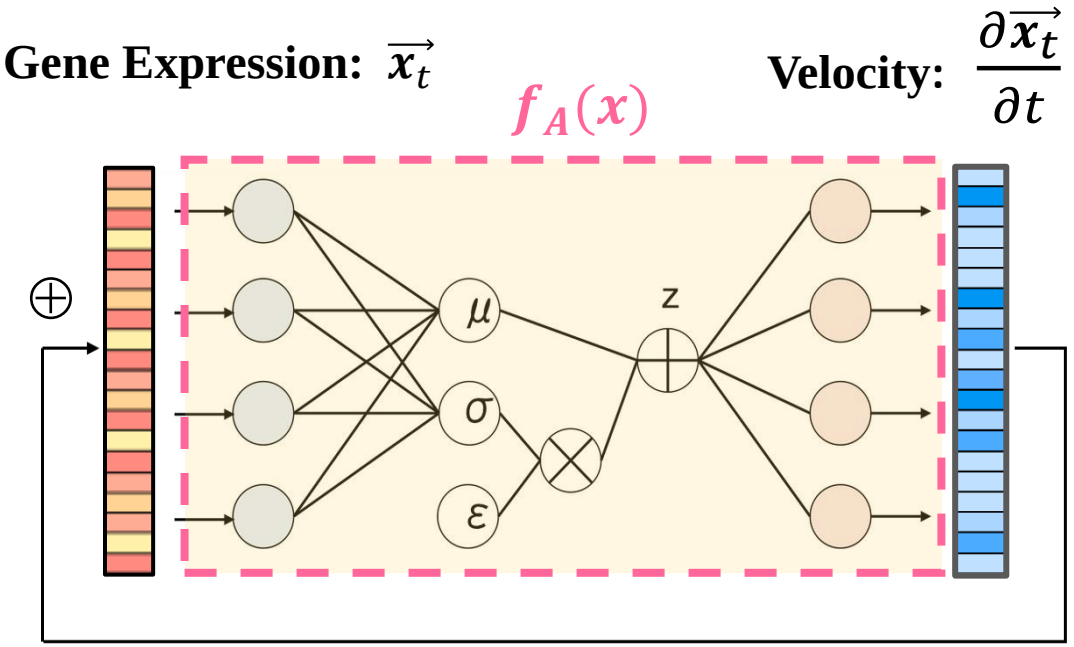
ODEnet

arXiv:1806.07366v5 [cs.LG] 14 Dec 2019

ODEnet I : Genome Science

Application to transcriptomics with VAE-joined ODE model

Gene Expression: $\overrightarrow{x_t}$ **Velocity:** $\frac{\partial \overrightarrow{x_t}}{\partial t}$



$$\frac{\partial \overrightarrow{x_t}}{\partial t} = f_A(\overrightarrow{x_t})$$

The first-order Euler's method for finding the state $\overrightarrow{x_{t+1}}$ is

$$\overrightarrow{x_{t+1}} = \overrightarrow{x_t} + f(\overrightarrow{x_t})$$

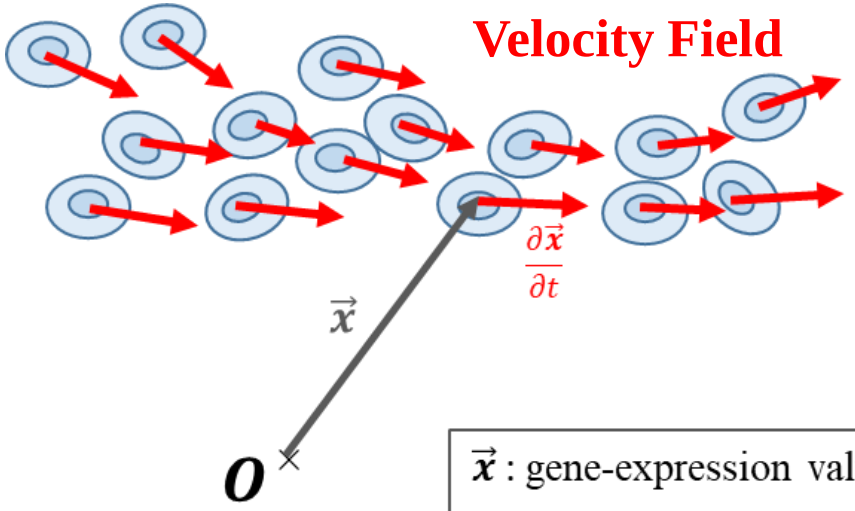
ODEs can be solved in the VAE-joined model.

Zhanlin Chen, William, C. King, Aheyon Hwang, Mark Gerstein, Jing Zhang, "**DeepVelo: Single-cell Transcriptomic Deep Velocity Field Learning with Neural Ordinary Differential Equations**", arXiv, 2022 https://doi.org/10.1101/2022.02.15.480564

細胞の状態空間における「場」を学習した後、そのAI（ODEnet）は細胞の状態を推論することができる！

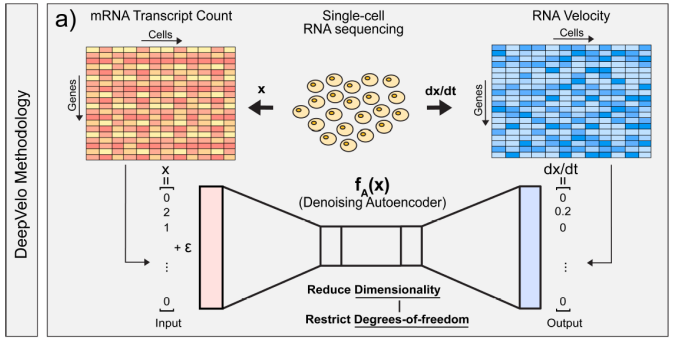
ODEnet I : Genome Science

Application to transcriptomics with VAE-joined ODE model



$\vec{x}$: gene-expression value (cell state)

$\frac{\partial \vec{x}}{\partial t}$: gene-expression velocity



Zhanlin Chen, William, C. King, Aheyon Hwang, Mark Gerstein, Jing Zhang, "**DeepVelo: Single-cell Transcriptomic Deep Velocity Field Learning with Neural Ordinary Differential Equations**", arXiv, 2022 https://doi.org/10.1101/2022.02.15.480564

By learning the velocity field in terms of the gene expression, a VAE-joined ODE model can predict the future cellular state across the time. This can be employed to explore the crucial factors that induce the specific cellular state in our analysis, and thereby it is expected to use the neural ODE model to identify target proteins that are responsible for a disease. This is based on an accurate, high quality prediction by ODEnet, whereas other AI models do not usually accomplish.

細胞の状態空間における「場」を学習した後、そのAI（ODEnet）は細胞の状態を推論することができる！

人工知能（AI）による創薬とは？

創薬のあらゆる領域において、 AI が既に活用されている！

今日のトピックス

分子設計（I）

タンパク質の**アミノ酸配列**（入力自体は立体構造ではない！）から、結合サイトとリガンドを Identify（Affinity）する！

分子設計（II）

タンパク質・化合物の複合体 **3D Structural Docking** Simulation

タンパク質と化合物のAffinity

医薬品の電子構造：　薬理効果、PK and PD、安全性、分子設計（III）

→ 様々な性質（Properties）を理解するための基礎

→ Screening and Reactivity の探索における基盤

創薬のためのターゲットの探索 → Genome Science の応用も！

Transcriptome, Proteome, Interactome, Epigenetics, etc.

Deep Learning of Wave Function to Construct a Precise XC Functional

Generation of ψ Coupled with Neural Networks

Kirkpatrick *et al.*, *Science* **374**, 1385–1389 (2021) 10 December 2021

QUANTUM CHEMISTRY

Pushing the frontiers of density functionals by solving the fractional electron problem

James Kirkpatrick[1]*†, Brendan McMorrow[1]†, David H. P. Turban[1]†, Alexander L. Gaunt[1]†, James S. Spencer[1], Alexander G. D. G. Matthews[1], Annette Obika[1], Louis Thiry[2], Meire Fortunato[1], David Pfau[1], Lara Román Castellanos[1], Stig Petersen[1], Alexander W. R. Nelson[1], Pushmeet Kohli[1], Paula Mori-Sánchez[3], Demis Hassabis[1], Aron J. Cohen[1,4]*

Density functional theory describes matter at the quantum level, but all popular approximations suffer from systematic errors that arise from the violation of mathematical properties of the exact functional. We overcame this fundamental limitation by training a neural network on molecular data and on fictitious systems with fractional charge and spin. The resulting functional, DM21 (DeepMind 21), correctly describes typical examples of artificial charge delocalization and strong correlation and performs better than traditional functionals on thorough benchmarks for main-group atoms and molecules. DM21 accurately models complex systems such as hydrogen chains, charged DNA base pairs, and diradical transition states. More crucially for the field, because our methodology relies on data and constraints, which are continually improving, it represents a viable pathway toward the exact universal functional.

[1]DeepMind, 6 Pancras Square, London N1C 4AG, UK. [2]Département d'informatique, ENS, CNRS, PSL University, Paris, France. [3]Departamento de Química and IFIMAC, UAM, 28049, Madrid, Spain. [4]Max Planck Institute for Solid State Research, 70569 Stuttgart, Germany.

Solving the fractional electron problem
(The delocalization error of electron density and The spin symmetry breaking)

In a DFT calculation, the functional determines the charge density of a molecule by finding the configuration of electrons which minimizes energy. Thus, errors in the functional can lead to errors in the calculated electron density.

And, when describing the breaking of chemical bonds, existing functionals tend to unrealistically prefer configurations in which a fundamental symmetry known as spin symmetry is broken.

DM21 predicts $E_{XC}[\rho(\vec{r})]$ as a E_{XC}^{MLP} with a Neural Network and incorporating exact properties into the training data.

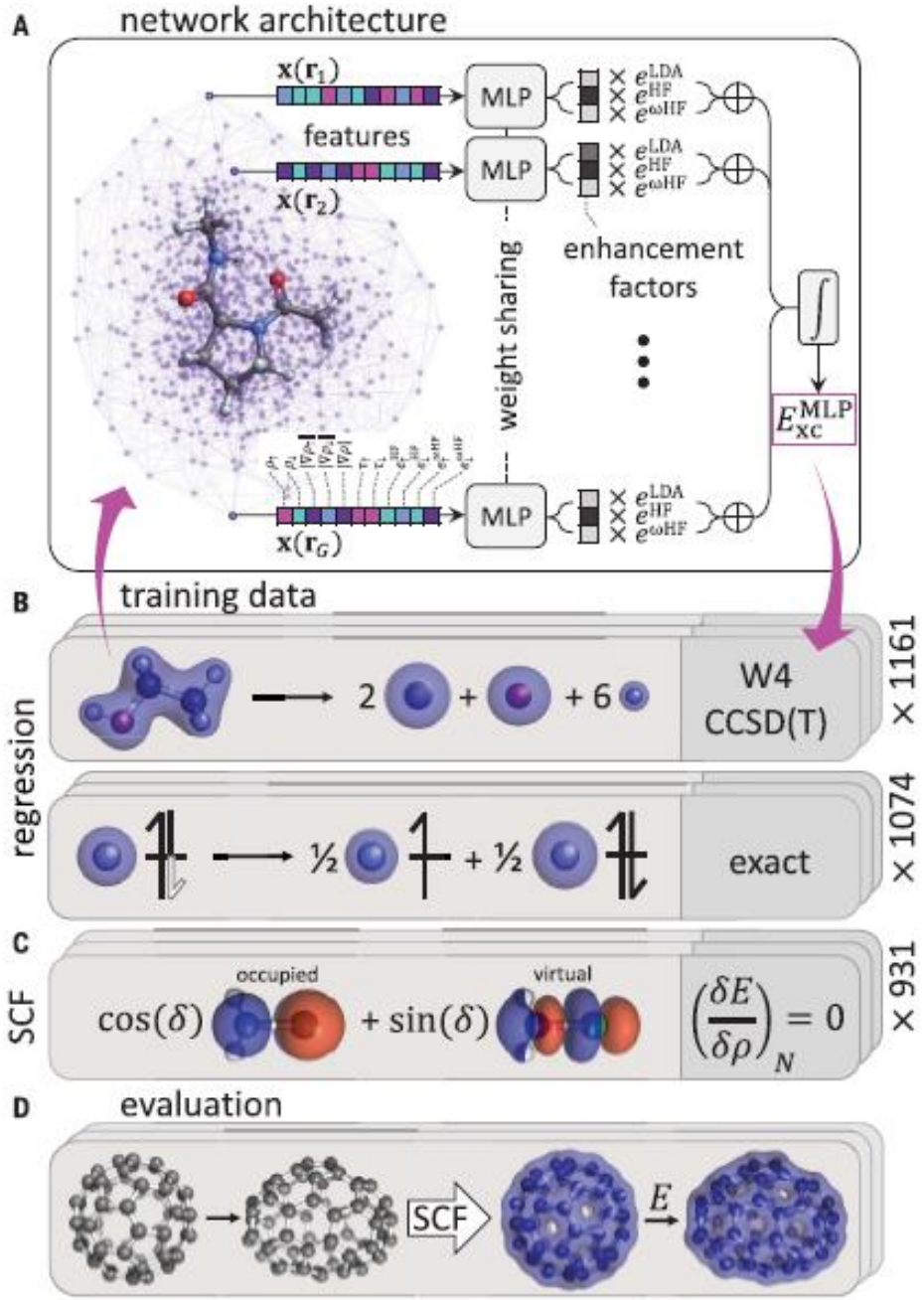
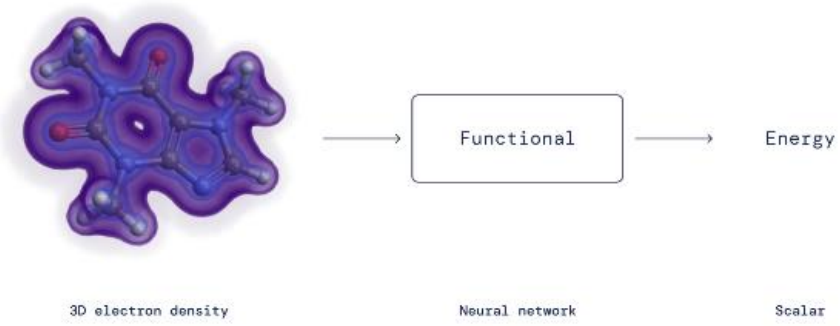


Fig. 1. Overview of the functional architecture and training. (**A**) Features of the electron density

Ref: WebSite of DeepMind https://www.deepmind.com/blog/simulating-matter-on-the-quantum-scale-with-ai

Deep Learning of Wave Function to Construct a Precise XC Functional



The followings are for you reference to get an overview of the present issues.

Nearly a century ago, **Erwin Schrodinger** propsed his famous equation govering the behavior of QM particles.
Then, **Pierre Hohenberg** and **Walter Kohn** realised that it is not necessary to track each electron individually, and instead, knowing the probability for any electron to be at each position (i.e., the electron density) is sufficient to exactly compute all interactions. After proving this, Kohn founded **Density Functional Theory (DFT)**.

$$E_{DFT}[\rho(\vec{r})] = T[\rho(\vec{r})] + V_{NE} + J[\rho(\vec{r})] + E_{XC}[\rho(\vec{r})]$$

The density functional has remained unknown and has to be approximated. By using a neural network to represent the exact functional, we found that we could solve the problems.

Handmade (approximation) → Neural Networks (exact functional)

These longstanding challenges are both related to how functionals behave when presented with a system that exhibits "fractional electron character." **By using a neural network to represent the functional and tailoring our training dataset to capture the fractional electron behavior expected for the exact functional, we found that we could solve the problems.**

Ref: WebSite of DeepMind https://www.deepmind.com/blog/simulating-matter-on-the-quantum-scale-with-ai

人工知能（AI）による創薬とは？

創薬のあらゆる領域において、 AI が既に活用されている！

どんな Fields で？

QM3D Descriptors of Compounds in Huge (Compound) Libraries
→ Use in Various Fields Relevant to DD Works

High Quality and Fast Molecular Dynamics (MD) Free Energy (FE) Calculation of Complexes of Protein and Ligand (for Screening of Compounds)

Transcriptome, Proteome, Epigenetics, Interactome, etc.

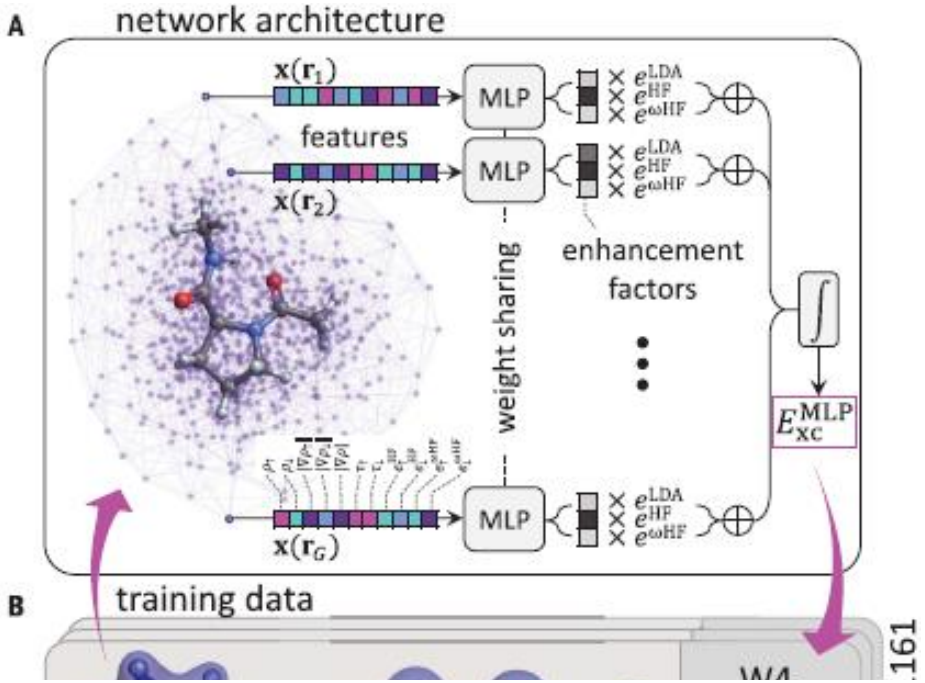
Protein-Protein Interaction (PPI)

医薬品の**生産技術**など ← Engineering

製造プロセス技術、製剤技術、(結晶) 構造解析技術、品質管理、etc.

Deep Learning of Wave Function to Construct a Precise XC Functional

Generation of ψ
Coupled with Neural Networks



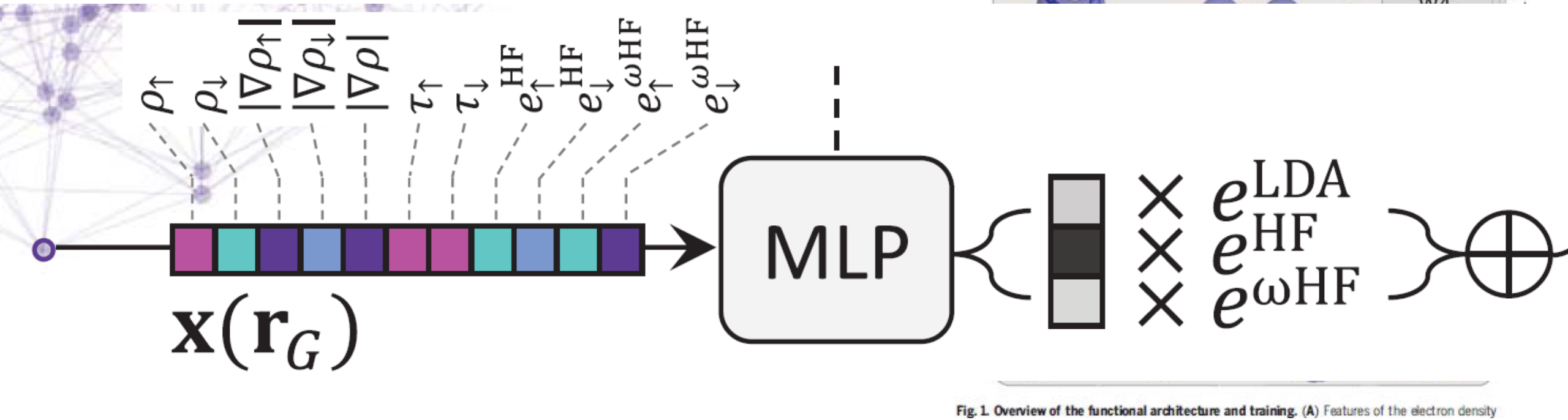


Fig. 1. Overview of the functional architecture and training. (A) Features of the electron density

Ref: WebSite of DeepMind https://www.deepmind.com/blog/simulating-matter-on-the-quantum-scale-with-ai

Deep Learning of Wave Function to Construct a Precise XC Functional

Kirkpatrick *et al.*, *Science* **374**, 1385–1389 (2021) 10 December 2021

QUANTUM CHEMISTRY

Pushing the frontiers of density functionals by solving the fractional electron problem

James Kirkpatrick[1]*†, Brendan McMorrow[1]†, David H. P. Turban[1]†, Alexander L. Gaunt[1]†, James S. Spencer[1], Alexander G. D. G. Matthews[1], Annette Obika[1], Louis Thiry[2], Meire Fortunato[1], David Pfau[1], Lara Román Castellanos[1], Stig Petersen[1], Alexander W. R. Nelson[1], Pushmeet Kohli[1], Paula Mori-Sánchez[3], Demis Hassabis[1], Aron J. Cohen[1,4]*

Density functional theory describes matter at the quantum level, but all popular approximations suffer from systematic errors that arise from the violation of mathematical properties of the exact functional. We overcame this fundamental limitation by training a neural network on molecular data and on fictitious systems with fractional charge and spin. The resulting functional, DM21 (DeepMind 21), correctly describes typical examples of artificial charge delocalization and strong correlation and performs better than traditional functionals on thorough benchmarks for main-grou atoms and molecules. DM21 accurately models complex systems such as hydrogen chains, charged DNA base pairs, and diradical transition states. More crucially for the field, because our methodology relies on data and constraints, which are continually improving, it represents a viable pathway toward the exact universal functional.

[1]DeepMind, 6 Pancras Square, London N1C 4AG, UK. [2]Département d'informatique, ENS, CNRS, PSL University, Paris, France. [3]Departamento de Química and IFIMAC, UAM, 28049, Madrid, Spain. [4]Max Planck Institute for Solid State Research, 70569 Stuttgart, Germany.

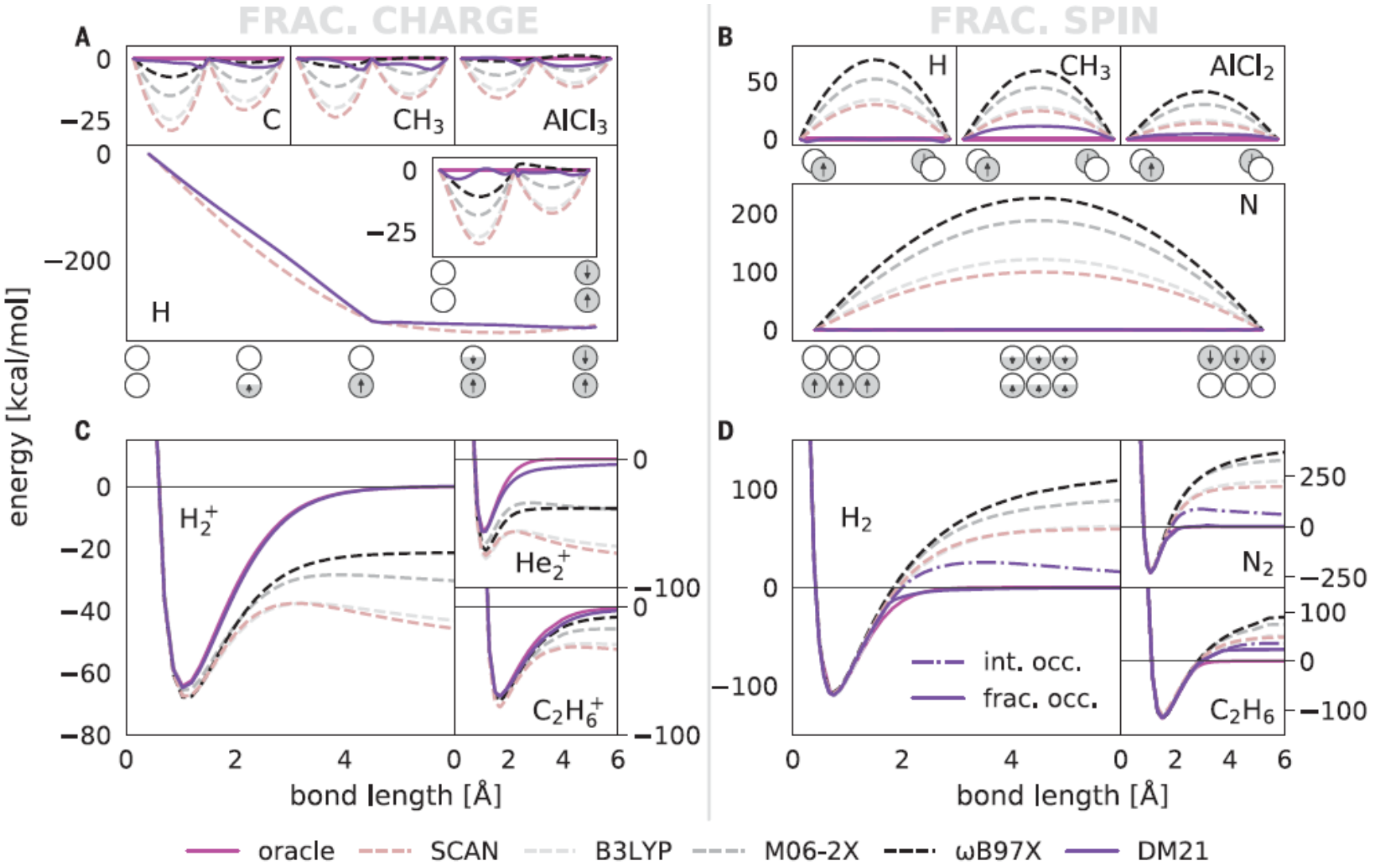


Fig. 2. Training on fractional electron constraints solves charge and spin generalizes to improved cation binding curves for DM21. The oracle is HF

Deep Learning of Wave Function to Construct a Precise XC Functional

Kirkpatrick *et al.*, *Science* **374**, 1385–1389 (2021) 10 December 2021

QUANTUM CHEMISTRY

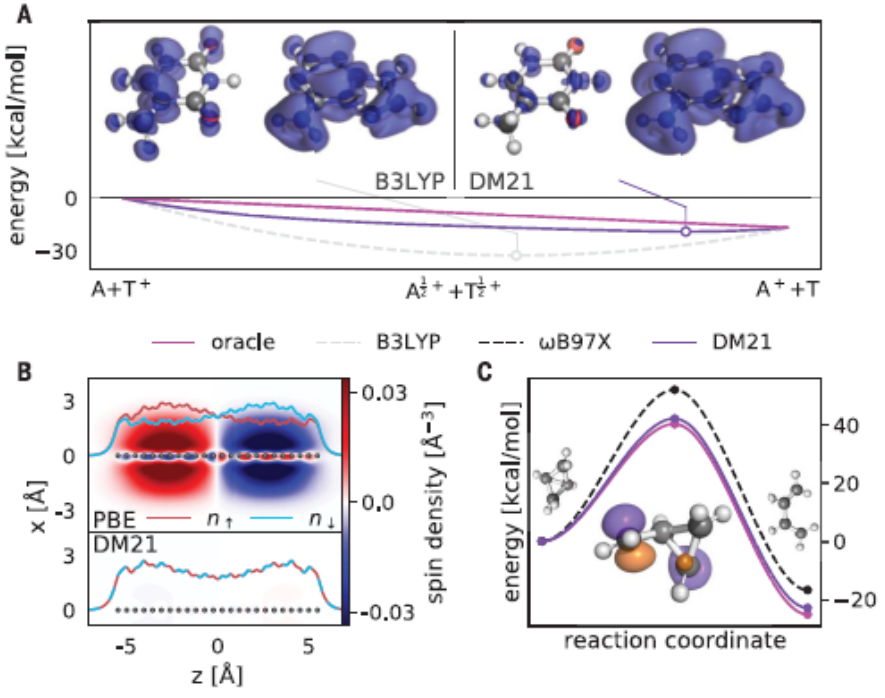
Pushing the frontiers of density functionals by solving the fractional electron problem

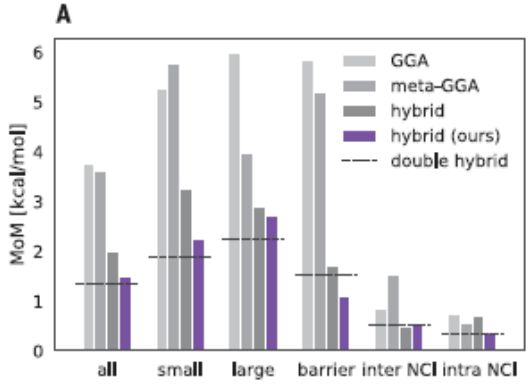
James Kirkpatrick[1*†], Brendan McMorrow[1†], David H. P. Turban[1†], Alexander L. Gaunt[1†], James S. Spencer[1], Alexander G. D. G. Matthews[1], Annette Obika[1], Louis Thiry[2], Meire Fortunato[1], David Pfau[1], Lara Román Castellanos[1], Stig Petersen[1], Alexander W. R. Nelson[1], Pushmeet Kohli[1], Paula Mori-Sánchez[3], Demis Hassabis[1], Aron J. Cohen[1,4*]

Density functional theory describes matter at the quantum level, but all popular approximations suffer from systematic errors that arise from the violation of mathematical properties of the exact functional. We overcame this fundamental limitation by training a neural network on molecular data and on fictitious systems with fractional charge and spin. The resulting functional, DM21 (DeepMind 21), correctly describes typical examples of artificial charge delocalization and strong correlation and performs better than traditional functionals on thorough benchmarks for main-group atoms and molecules. DM21 accurately models complex systems such as hydrogen chains, charged DNA base pairs, and diradical transition states. More crucially for the field, because our methodology relies on data and constraints, which are continually improving, it represents a viable pathway toward the exact universal functional.

[1]DeepMind, 6 Pancras Square, London N1C 4AG, UK. [2]Département d'informatique, ENS, CNRS, PSL University, Paris, France. [3]Departamento de Química and IFIMAC, UAM, 28049, Madrid, Spain. [4]Max Planck Institute for Solid State Research, 70569 Stuttgart, Germany.

Fig. 3. Exact constraints improve performance on challenging chemistry. (**A**) Charge density for a singly ionized adenine-thymine base pair. B3LYP unphysically delocalizes charge on both base pairs despite adenine having a deeper ionization potential. Conversely, DM21 displays localization of charge on the adenine only. (**B**) Spin density for a compressed chain of 24 hydrogen atoms at 0.48 Å separation. The line density n for each spin channel is overlaid (supplementary materials, section 3.3). The PBE functional (*41*) breaks spin symmetry and leads to an apparent magnetization along the chain. This effect is also predicted by other functionals but is absent in DM21. (**C**) The conrotatory pathways of bicyclobutane isomerization. The HOMO of a single spin channel in an unrestricted calculation is shown for the transition states. Spin is delocalized across two atoms in the conrotatory path, requiring satisfaction of FS for accurate modeling. The oracle is diffusion Monte Carlo from (*42*).





B

	GMTKN55 MoM	BBB	QM9
SCAN(:D3$_{BJ}$)	3.63	52.90	3.57
ωB97X(-V)	2.45	70.28	2.13
M06-2X	2.08	64.50	2.20
PW6B95:D3$_0$	1.98	57.81	2.24
DM21	**1.50**	**13.20**	**1.66**

Fig. 4. State-of-the-art performance by DM21 on benchmarks. All errors

benchmark sets: BBB (supplementary materials, section 8.1), GMTKN55,

人工知能（AI）による創薬とは？

創薬のあらゆる領域において、 AI が既に活用されている！

今日のトピックス

分子設計（I）

タンパク質の**アミノ酸配列**（入力自体は立体構造ではない！）から、結合サイトとリガンドを Identify（Affinity）する！

分子設計（II）

タンパク質・化合物の複合体 **3D Structural Docking** Simulation

タンパク質と化合物のAffinity

医薬品の電子構造：　薬理効果、PK and PD、安全性、分子設計（III）

→ 様々な性質（Properties）を理解するための基礎

→ Screening and Reactivity の探索における基盤

創薬のためのターゲットの探索 → Genome Science の応用も！

Transcriptome, Proteome, Interactome, Epigenetics, etc.

Manifold-Based Geometrical Techniques

Manifold Embedding of Biological Multi-Modal Data

Multi Modal Data

Patch Clump and RNA-Seq

Article | Published: 31 January 2022

A deep manifold-regularized learning model for improving phenotype prediction from multi-modal data

Nam D. Nguyen, Jiawei Huang & Daifeng Wang

Nature Computational Science **2**, 38–46 (2022) | Cite this article

Aim :
a deeper understanding of underlying complex mechanisms across scales for phenotypes

Method :
an interpretable regularized learning model, deepManReg

Single Cell Multi-Modal Data (**Patch-Seq** Data)
including **transcriptomics** and **electrophysiology**
for neuronal cells in the mouse brain

Manifold-Based Geometrical Techniques

Data source: multi-modal data from mouse brain neuronal cells

Article

Integrated Morphoelectric and Transcriptomic Classification of Cortical GABAergic Cells

Nathan W. Gouwens,[1,4,*] Staci A. Sorensen,[1,4,*] Fahimeh Baftizadeh,[1] Agata Budzillo,[1] Brian R. Lee,[1] Tim Jarsky,[1] Lauren Alfiler,[1] Katherine Baker,[1] Eliza Barkan,[1] Kyla Berry,[1] Darren Bertagnolli,[1] Kris Bickley,[1] Jasmine Bomben,[1] Thomas Braun,[2] Krissy Brouner,[1] Tamara Casper,[1] Kirsten Crichton,[1] Tanya L. Daigle,[1] Rachel Dalley,[1] Rebecca A. de Frates,[1] Nick Dee,[1] Tsega Desta,[1] Samuel Dingman Lee,[1] Nadezhda Dotson,[1] Tom Egdorf,[1] Lauren Ellingwood,[1] Rachel Enstrom,[1] Luke Esposito,[1] Colin Farrell,[1] David Feng,[1] Olivia Fong,[1] Rohan Gala,[1] Clare Gamlin,[1] Amanda Gary,[1] Alexandra Glandon,[1] Jeff Goldy,[1] Melissa Gorham,[1] Lucas Graybuck,[1] Hong Gu,[1] Kristen Hadley,[1] Michael J. Hawrylycz,[1] Alex M. Henry,[1] DiJon Hill,[1] Madie Hupp,[1] Sara Kebede,[1] Tae Kyung Kim,[1] Lisa Kim,[1] Matthew Kroll,[1] Changkyu Lee,[1] Katherine E. Link,[1] Matthew Mallory,[1] Rusty Mann,[1] Michelle Maxwell,[1] Medea McGraw,[1] Delissa McMillen,[1] Alice Mukora,[1] Lindsay Ng,[1] Lydia Ng,[1] Kiet Ngo,[1] Philip R. Nicovich,[1] Aaron Oldre,[1] Daniel Park,[1] Hanchuan Peng,[1] Osnat Penn,[1] Thanh Pham,[1] Alice Pom,[1] Zoran Popović,[3] Lydia Potekhina,[1] Ramkumar Rajanbabu,[1] Shea Ransford,[1] David Reid,[1] Christine Rimorin,[1] Miranda Robertson,[1] Kara Ronellenfitch,[1] Augustin Ruiz,[1] David Sandman,[1] Kimberly Smith,[1] Josef Sulc,[1] Susan M. Sunkin,[1] Aaron Szafer,[1] Michael Tieu,[1] Amy Torkelson,[1] Jessica Trinh,[1] Herman Tung,[1] Wayne Wakeman,[1] Katelyn Ward,[1] Grace Williams,[1] Zhi Zhou,[1] Jonathan T. Ting,[1] Anton Arkhipov,[1] Uygar Sümbül,[1] Ed S. Lein,[1] Christof Koch,[1] Zizhen Yao,[1] Bosiljka Tasic,[1] Jim Berg,[1] Gabe J. Murphy,[1,5,*] and Hongkui Zeng[1]

[1]Allen Institute for Brain Science, Seattle, WA 98109, USA
[2]Byte Physics, Schwarzastraße 9, Berlin 12055, Germany
[3]Center for Game Science, Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA 98195, USA
[4]These authors contributed equally
[5]Lead Contact

Cell *183*, 935–953, November 12, 2020

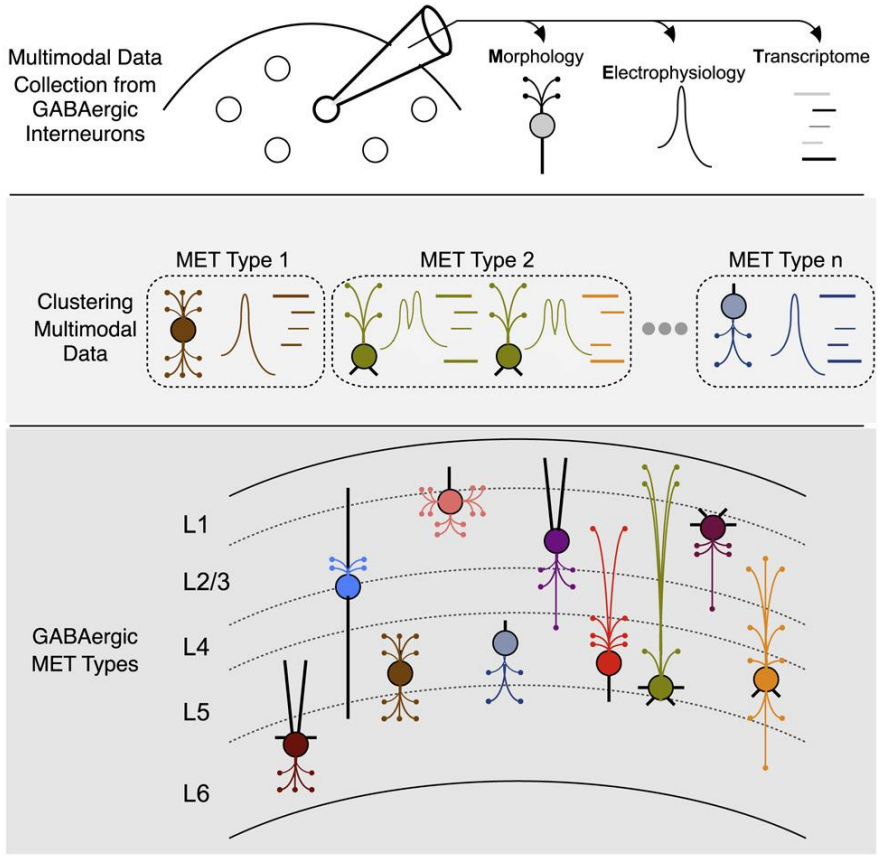
Mouse visual cortex
(4020 neurons)

- Gene expression
- Electrophysiology

Multi-modal feature

Classification

(brain layer and electrophysiological property)



Findings:

Neurons can be defined into **28** types

Distinguished by their layer-specific connection patterns

Curved Space（曲がった空間）

曲がった空間における平行移動は可能か？

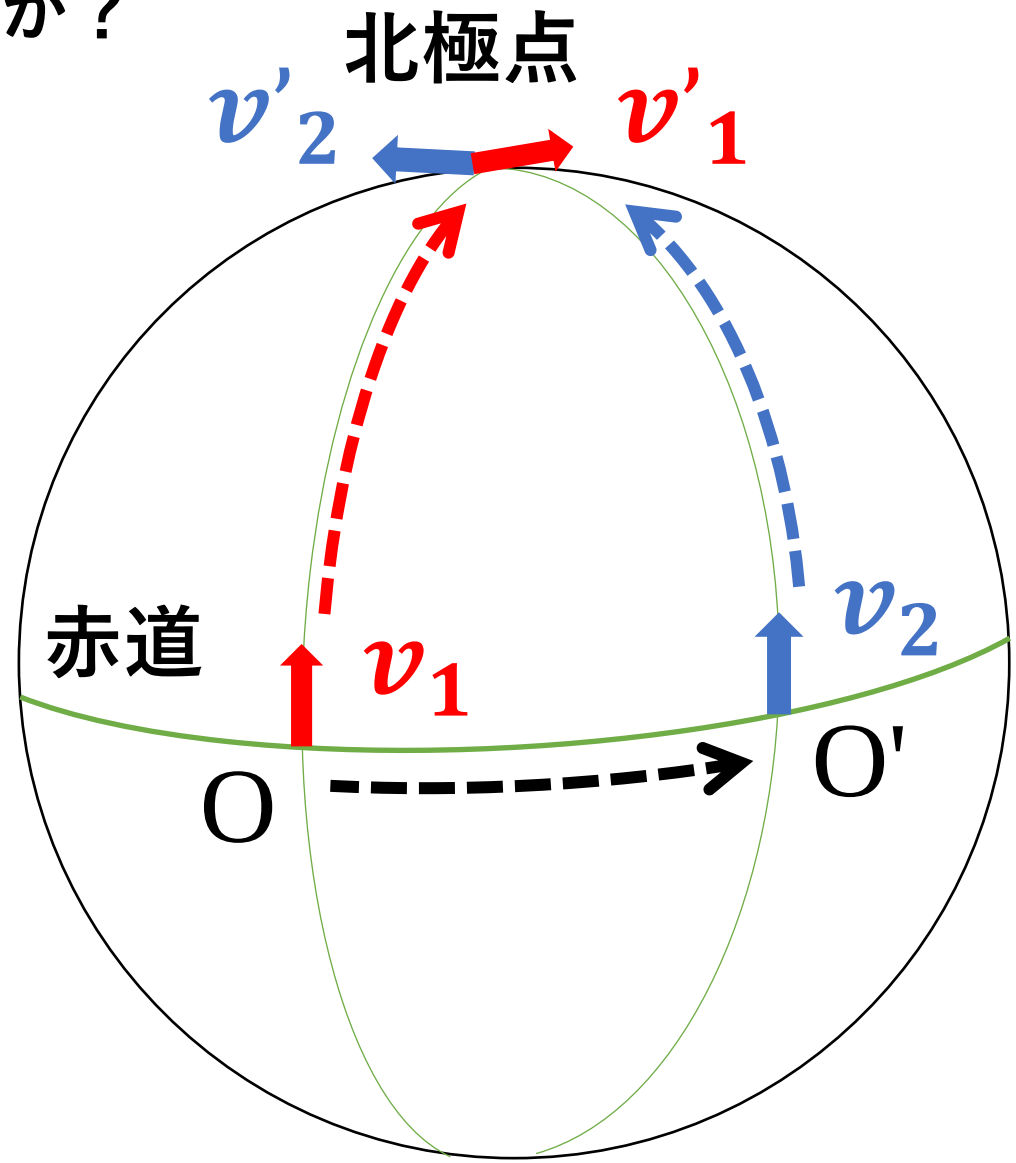
地球の表面において，

v_1：赤道上の点 O における、
経線方向の接線ベクトル

v_2：赤道上を移動し、点 O' から
経線方向の接線ベクトル

これらのふたつの接線ベクトル v'_1，v'_2 が北極点に移動した後に、それらのベクトルの向きを比較すると、大きく異なる！

→ 空間の**曲率** → **曲がった空間**！



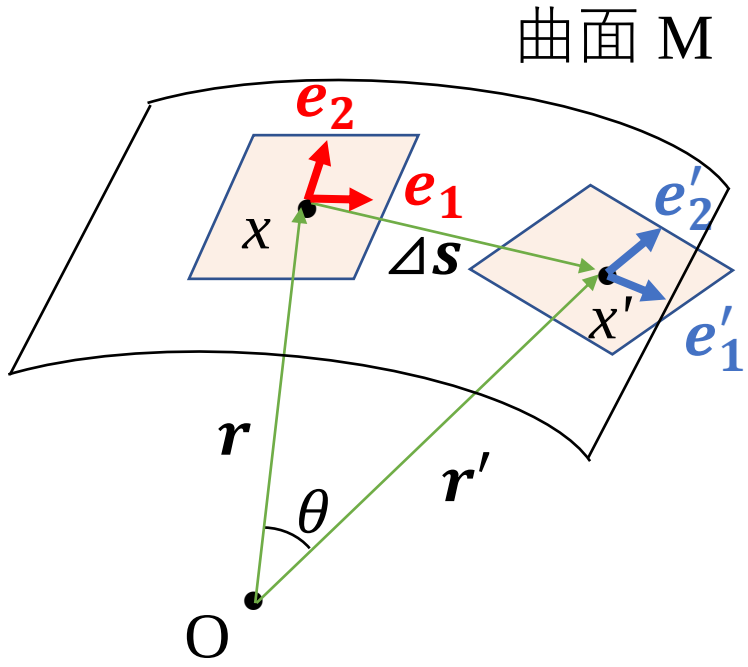
Curved Space（曲がった空間）

曲がった空間における微分は可能か？

Tangent Plane（**接平面**）： $T_x(\mathrm{M})$
　曲面 M における、ある点 x での**接平面**

Tangent Space（**接空間**）： TM
　Tangent Planes の集合

Tangent Vector Bundle（**接ベクトル バンドル**）



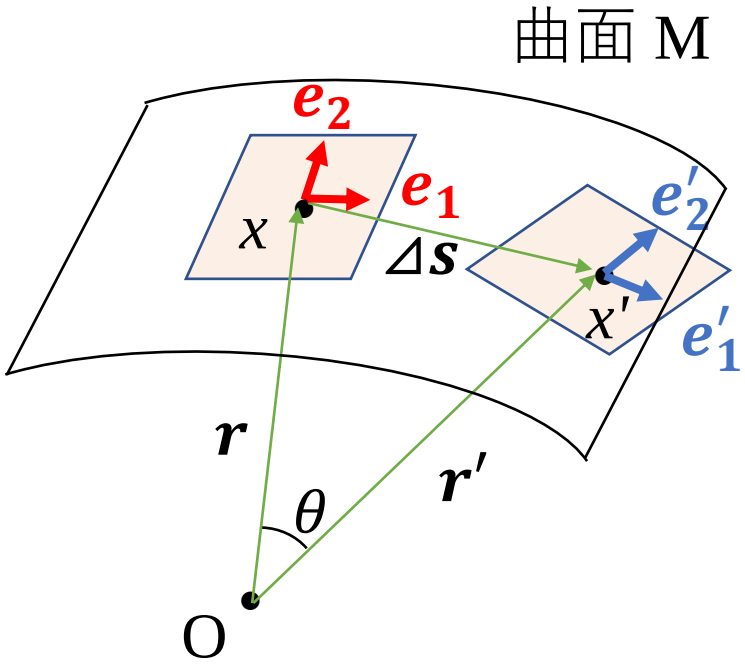
e_1, e_2 ：点 x における Tangent Plane の Basis Set

e'_1, e'_2 ：点 x' における Tangent Plane の Basis Set

Curved Space（曲がった空間）

曲がった空間における微分は可能か？

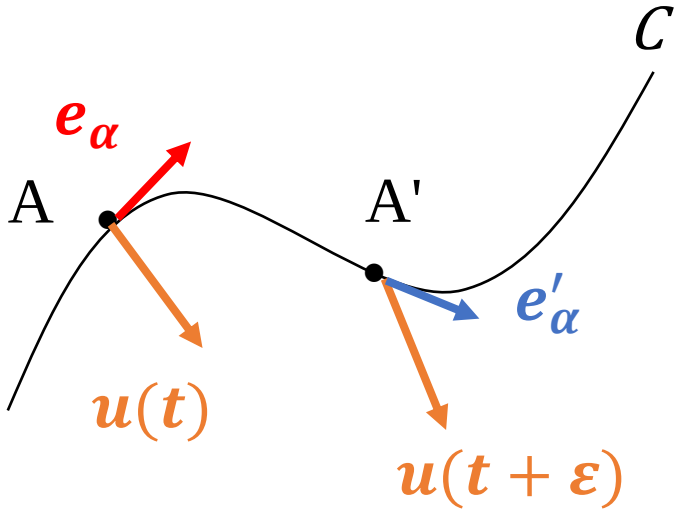
曲面 M における Vector Field $\boldsymbol{u}$ において、曲線C上の位置がパラメータ t によって決められているとし、これを $\boldsymbol{u}(t)$ と記す。



ここで $\boldsymbol{u}(t)$ を Tangent Plane の Basis Set によって展開すると、

$$\boldsymbol{u}(t) = \sum_{\mu} u^{\mu}\, \boldsymbol{e}_{\mu} = u^{\mu}\, \boldsymbol{e}_{\mu}$$

と書ける。ただし右辺は、**Einstein's Summation Convention**を用いた（→ **Suffices が上下に揃う場合には、Summation の記号 Σ を略する**）。



Curved Space（曲がった空間）

曲がった空間における微分は可能か？

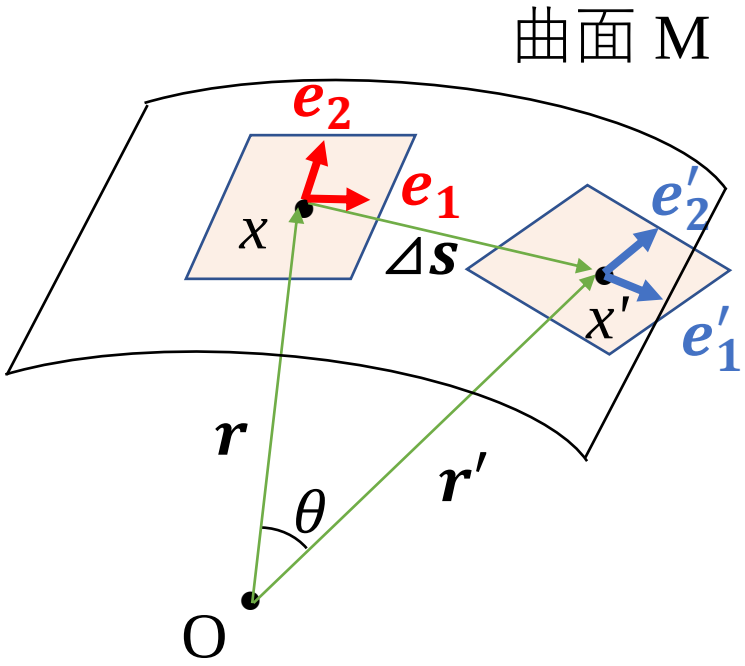
曲がった空間における Basis $\boldsymbol{e_\mu}$ の（$\boldsymbol{e_\alpha}$ の方向への）微分は、

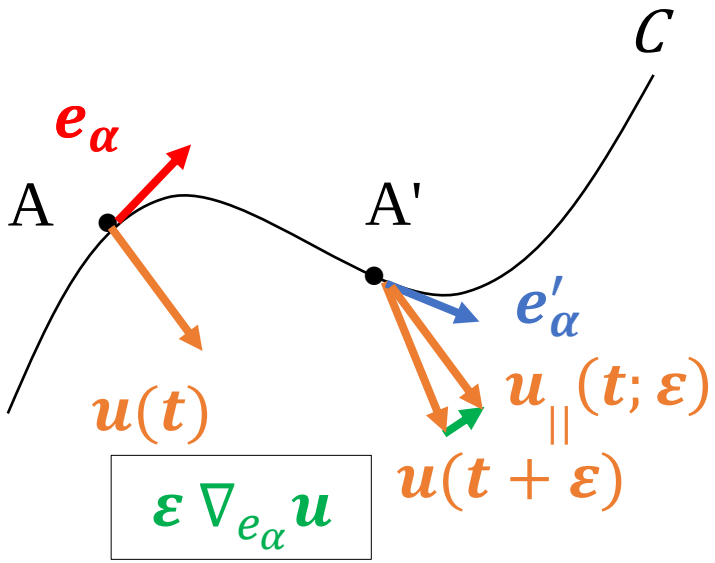
$$\nabla_{e_\alpha} \boldsymbol{e_\mu} = \lim_{\varepsilon \to 0} \frac{\boldsymbol{e_\mu}(t+\varepsilon) - \boldsymbol{e_{\mu\,||}}(t;\varepsilon)}{\varepsilon}$$

と書ける。よって、左図（$u(t)$ の微分を示す）において緑色の無限小ベクトルをBasis $\boldsymbol{e_\mu}$ に応用し、

$$\varepsilon \nabla_{e_\alpha} \boldsymbol{e_\mu}$$

と与えられる。微分は Tangent Space 上に存在する。



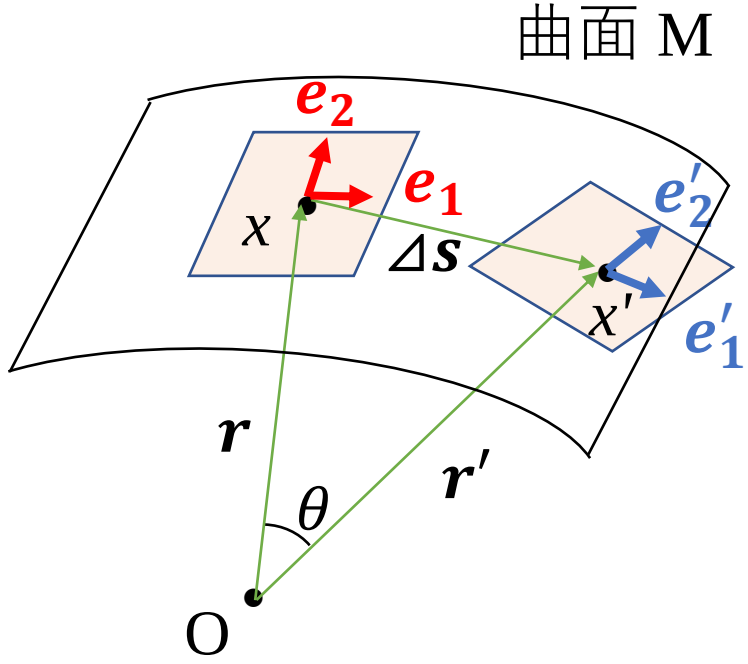


Curved Space（曲がった空間）

曲がった空間における微分は可能か？

（再掲） Vector Field $\boldsymbol{u}(t)$

$$\boldsymbol{u}(t) = \sum_{\mu} u^{\mu}\, \boldsymbol{e_{\mu}} = u^{\mu}\, \boldsymbol{e_{\mu}}$$

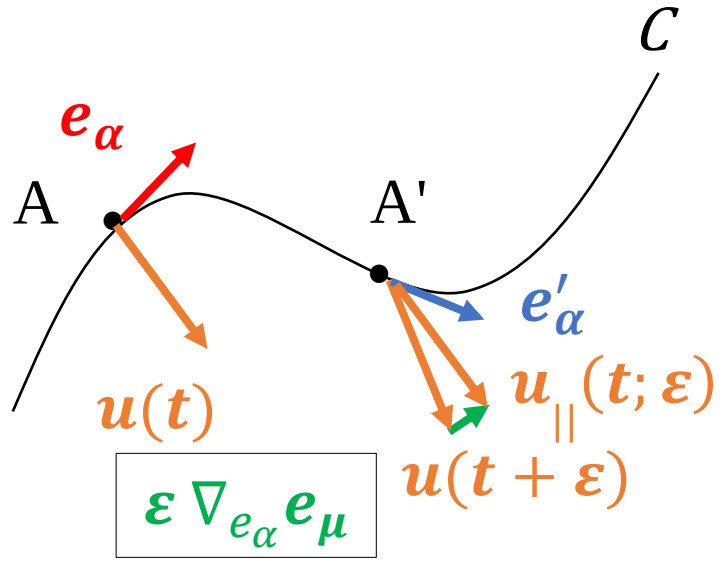


このとき $\boldsymbol{u}(t)$ の微分は、Leibnitz's Rule (Chain Rule) により

$$\nabla_{e_\alpha} u(t) = \nabla_{e_\alpha}\left(u^{\mu}\, \boldsymbol{e}_{\mu}\right)$$

$$= (\partial_{\alpha} u^{\mu})\boldsymbol{e_{\mu}} + u^{\mu}(\nabla_{e_\alpha} \boldsymbol{e}_{\mu})$$

を得る。ここで第２項も Basis Set $\{\boldsymbol{e}_{\beta}\}$ で展開し、



Curved Space（曲がった空間）

曲がった空間における微分は可能か？

そのために $\boldsymbol{e}_\beta$ の係数を $\Gamma^{\mu}_{\beta a}$ とおくと、

$$\nabla_{e_\alpha}\boldsymbol{e}_\mu = \Gamma^{\beta}_{\mu\alpha}\,\boldsymbol{e}_\beta$$

と書ける。よって、

$$\nabla_{e_\alpha}u(t) = \nabla_{e_\alpha}\left(u^\mu\,\boldsymbol{e}_\mu\right)$$

$$= (\partial_\alpha u^\mu)\boldsymbol{e}_{\boldsymbol{\mu}} + u^\mu(\nabla_{e_\alpha}\boldsymbol{e}_\mu)$$

となるから、ここで上式を代入し、

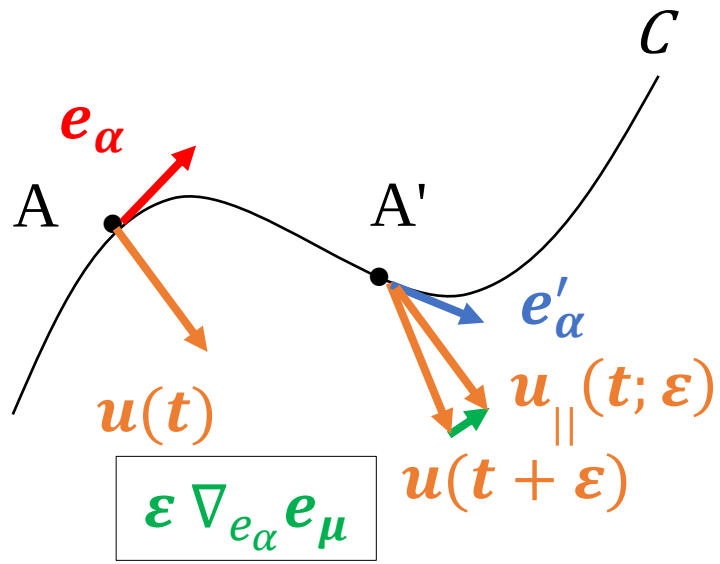
$$= (\partial_\alpha u^\mu)\boldsymbol{e}_{\boldsymbol{\mu}} + u^\beta\,\Gamma^{\mu}_{\beta\alpha}\boldsymbol{e}_\mu$$

となり、整理すると

$$= (\partial_\alpha u^\mu + u^\beta\,\Gamma^{\mu}_{\beta\alpha})\,\boldsymbol{e}_\mu$$

を得る。

定義により明らかなように $\Gamma^{\mu}_{\beta a}$ は本来 $\Gamma^{\mu}{}_{\beta a}$ と書くべきであるが、スペースの節約のために以下では左のように表記する。



Curved Space（曲がった空間）

曲がった空間における微分は可能か？

これを整理すると結局のところ、

$$\nabla_{e_\alpha} u(t) = \nabla_{e_\alpha}\left(u^\mu \boldsymbol{e}_\mu\right)$$

$$= \left(\partial_\alpha u^\mu + u^\beta \Gamma^\mu_{\beta\alpha}\right) \boldsymbol{e}_\mu$$

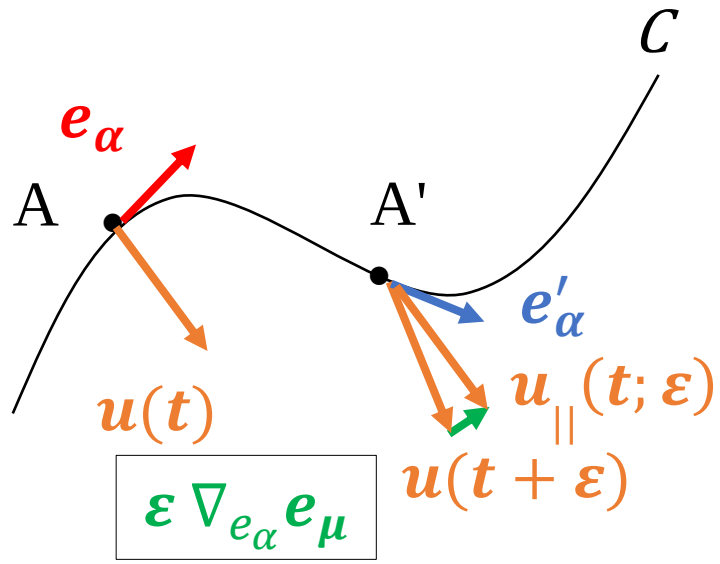
ここで　(definition) $= u^\mu_{;\alpha} \boldsymbol{e}_\mu$

と書く。すなわち、微分の成分は

$$\text{(definition)}\ u^\mu_{;\alpha} = \partial_\alpha u^\mu + u^\beta \Gamma^\mu_{\beta\alpha}$$

によって与えられる。これを特に **Covariance Differentiation**（共変微分）という。

定義により明らかなように $\Gamma^\mu_{\beta a}$ は本来 $\Gamma^\mu{}_{\beta a}$ と書くべきであるが、スペースの節約のために以下では左のように表記する。



Curved Space（曲がった空間）

曲がった空間における微分は可能か？

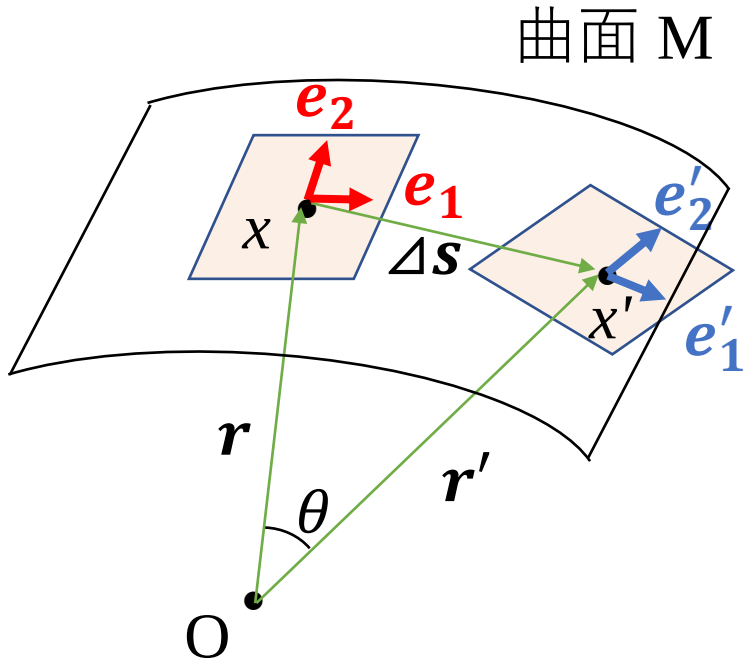
Vector Field $\boldsymbol{u}(t)$

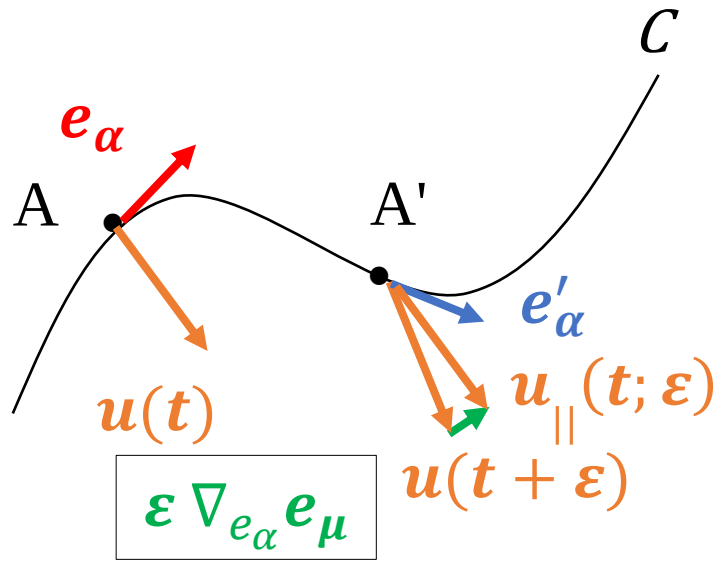
$$\boldsymbol{u}(t) = \sum_{\mu} u^{\mu}\, \boldsymbol{e}_{\boldsymbol{\mu}} = u^{\mu}\, \boldsymbol{e}_{\boldsymbol{\mu}}$$

$$\nabla_{e_\alpha} u(t) = \nabla_{e_\alpha}\left(u^{\mu}\, \boldsymbol{e}_{\mu}\right)$$

$$= (\partial_\alpha u^{\mu})\boldsymbol{e}_{\boldsymbol{\mu}} + (u^{\beta}\, \Gamma^{\mu}_{\beta\alpha})\boldsymbol{e}_{\mu}$$

$\boldsymbol{u}(t)$ のこの形の微分を特に**共変微分**という。上式に Leibnitz' Rule (Chain Rule) をapply すればよい。このように曲がった空間における微分は、通常の微分に「**Basis Set に対する微分**」を加えることで得られる。

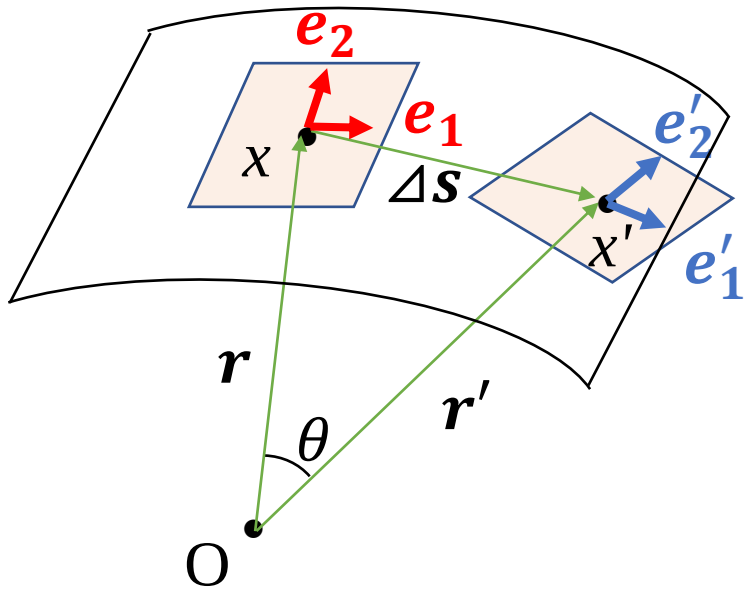




Curved Space（曲がった空間）

曲がった空間における微分は可能か？

曲面 M



Vector Field $\boldsymbol{u}(t)$

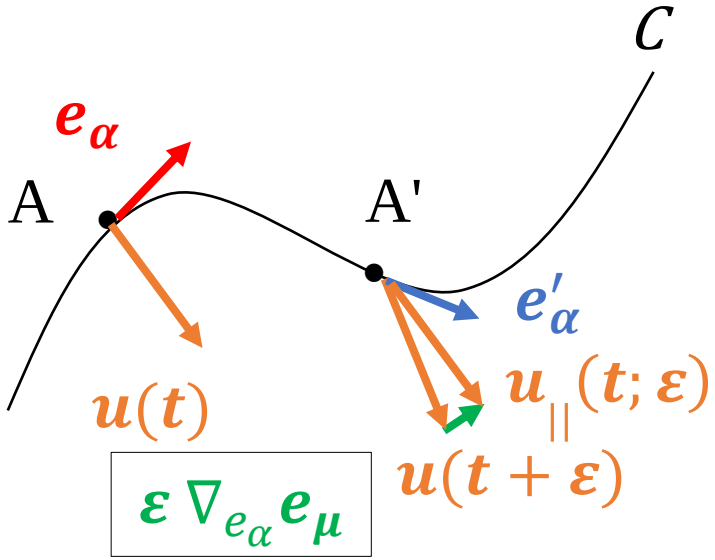
$$\boldsymbol{u}(t) = \sum_{\mu} u^{\mu}\, \boldsymbol{e_{\mu}} = u^{\mu}\, \boldsymbol{e_{\mu}}$$

$$\nabla_{e_\alpha} u(t) = (\partial_\alpha u^{\mu})\boldsymbol{e_{\mu}} + u^{\beta}\, \Gamma^{\mu}_{\beta\alpha} \boldsymbol{e}_{\mu}$$

これは以下のスローガンに整理できる！

$$\nabla_{e_\alpha} u(t) = \quad \partial \quad + \quad \Gamma$$

（ 普通の微分） ＋（Basis Set の微分）



Curved Space（曲がった空間）

曲がった空間における平行移動は可能か？

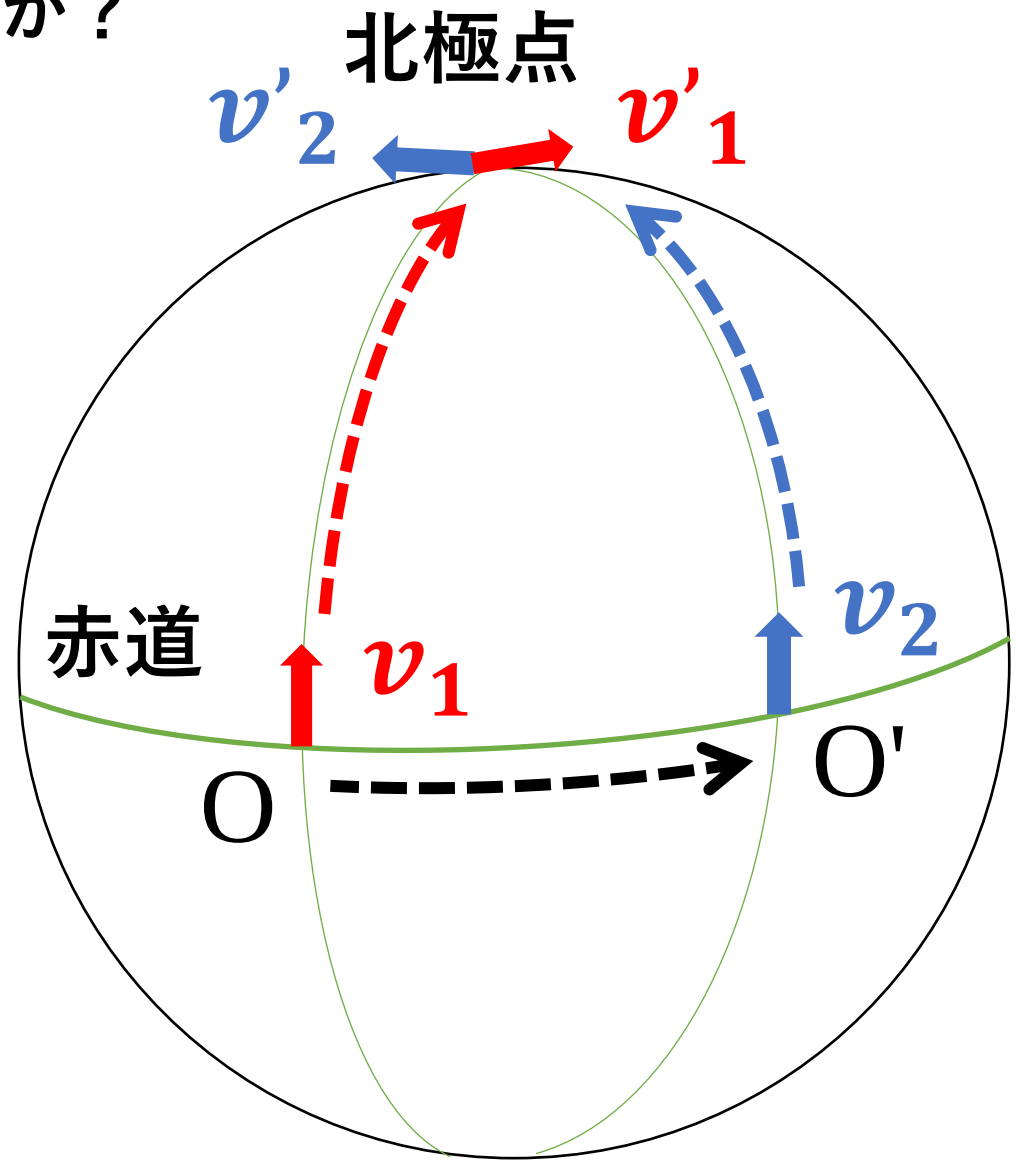
地球の表面において，

v_1：赤道上の点 O における、
経線方向の接線ベクトル

v_2：赤道上を移動し、点 O' から
経線方向の接線ベクトル

これらのふたつの接線ベクトル v'_1，v'_2 が北極点に移動した後に、それらのベクトルの向きを比較すると、大きく異なる！

→ 空間の**曲率** → **曲がった空間**！



Curved Space（曲がった空間）

曲がった空間における微分は可能か？

$\Gamma^{\mu}_{\beta\alpha}$ ： **接続係数**（Christoffel Symbol）

Riemann 接続： Basis Set 間の平行移動を Christoffel Symbol によって定義する（Riemann 多様体）。

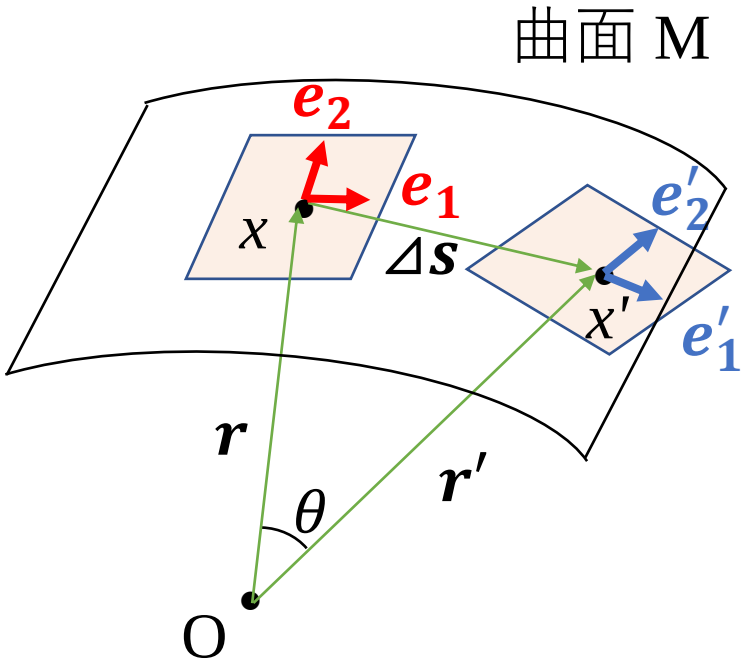
・前述のように、**経路によってベクトルの平行移動の結果は異なる**。それに対応して共変微分もまた**非可換**である。これは空間の**曲率**によるものである。

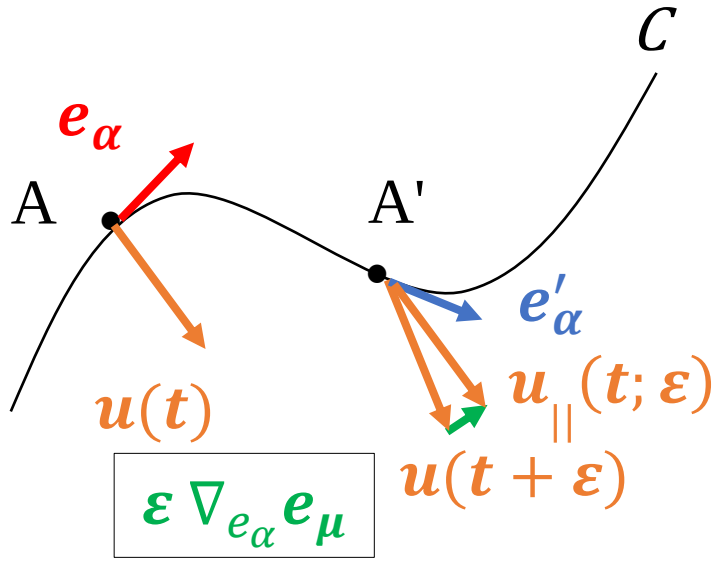
・そこで閉じた経路を１周したときに生じる、ベクトルの平行移動の差によって曲率テンソルを定義する（**Riemann Curvature Tensor**）。

注）Christoffel Symbol 自体は Tensor ではないが、例えば接続係数の差（変分など）を考えると Tensor となり得る。

Ex)
　　電磁場におけるゲージ場は接続に相当し、具体的には（電磁場における）Vector Potential A^{μ} が接続を構成する。

このようにして、曲がった空間における微分や曲率を定義できる！





Manifold-Based Geometrical Techniques

Manifold Embedding of Biological Multi-Modal Data

Multi Modal Data

Patch Clump and RNA-Seq

Article | Published: 31 January 2022

A deep manifold-regularized learning model for improving phenotype prediction from multi-modal data

Nam D. Nguyen, Jiawei Huang & Daifeng Wang

Nature Computational Science **2**, 38–46 (2022) | Cite this article

Aim :
a deeper understanding of underlying complex mechanisms across scales for phenotypes

Method :
an interpretable regularized learning model, deepManReg

Single Cell Multi-Modal Data (**Patch-Seq** Data)
including **transcriptomics** and **electrophysiology**
for neuronal cells in the mouse brain

Manifold-Based Geometrical Techniques

Data source: multi-modal data from mouse brain neuronal cells

Article

Integrated Morphoelectric and Transcriptomic Classification of Cortical GABAergic Cells

Nathan W. Gouwens,[1,4,*] Staci A. Sorensen,[1,4,*] Fahimeh Baftizadeh,[1] Agata Budzillo,[1] Brian R. Lee,[1] Tim Jarsky,[1] Lauren Alfiler,[1] Katherine Baker,[1] Eliza Barkan,[1] Kyla Berry,[1] Darren Bertagnolli,[1] Kris Bickley,[1] Jasmine Bomben,[1] Thomas Braun,[2] Krissy Brouner,[1] Tamara Casper,[1] Kirsten Crichton,[1] Tanya L. Daigle,[1] Rachel Dalley,[1] Rebecca A. de Frates,[1] Nick Dee,[1] Tsega Desta,[1] Samuel Dingman Lee,[1] Nadezhda Dotson,[1] Tom Egdorf,[1] Lauren Ellingwood,[1] Rachel Enstrom,[1] Luke Esposito,[1] Colin Farrell,[1] David Feng,[1] Olivia Fong,[1] Rohan Gala,[1] Clare Gamlin,[1] Amanda Gary,[1] Alexandra Glandon,[1] Jeff Goldy,[1] Melissa Gorham,[1] Lucas Graybuck,[1] Hong Gu,[1] Kristen Hadley,[1] Michael J. Hawrylycz,[1] Alex M. Henry,[1] DiJon Hill,[1] Madie Hupp,[1] Sara Kebede,[1] Tae Kyung Kim,[1] Lisa Kim,[1] Matthew Kroll,[1] Changkyu Lee,[1] Katherine E. Link,[1] Matthew Mallory,[1] Rusty Mann,[1] Michelle Maxwell,[1] Medea McGraw,[1] Delissa McMillen,[1] Alice Mukora,[1] Lindsay Ng,[1] Lydia Ng,[1] Kiet Ngo,[1] Philip R. Nicovich,[1] Aaron Oldre,[1] Daniel Park,[1] Hanchuan Peng,[1] Osnat Penn,[1] Thanh Pham,[1] Alice Pom,[1] Zoran Popović,[3] Lydia Potekhina,[1] Ramkumar Rajanbabu,[1] Shea Ransford,[1] David Reid,[1] Christine Rimorin,[1] Miranda Robertson,[1] Kara Ronellenfitch,[1] Augustin Ruiz,[1] David Sandman,[1] Kimberly Smith,[1] Josef Sulc,[1] Susan M. Sunkin,[1] Aaron Szafer,[1] Michael Tieu,[1] Amy Torkelson,[1] Jessica Trinh,[1] Herman Tung,[1] Wayne Wakeman,[1] Katelyn Ward,[1] Grace Williams,[1] Zhi Zhou,[1] Jonathan T. Ting,[1] Anton Arkhipov,[1] Uygar Sümbül,[1] Ed S. Lein,[1] Christof Koch,[1] Zizhen Yao,[1] Bosiljka Tasic,[1] Jim Berg,[1] Gabe J. Murphy,[1,5,*] and Hongkui Zeng[1]

[1]Allen Institute for Brain Science, Seattle, WA 98109, USA
[2]Byte Physics, Schwarzastraße 9, Berlin 12055, Germany
[3]Center for Game Science, Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA 98195, USA
[4]These authors contributed equally
[5]Lead Contact

Cell *183*, 935–953, November 12, 2020

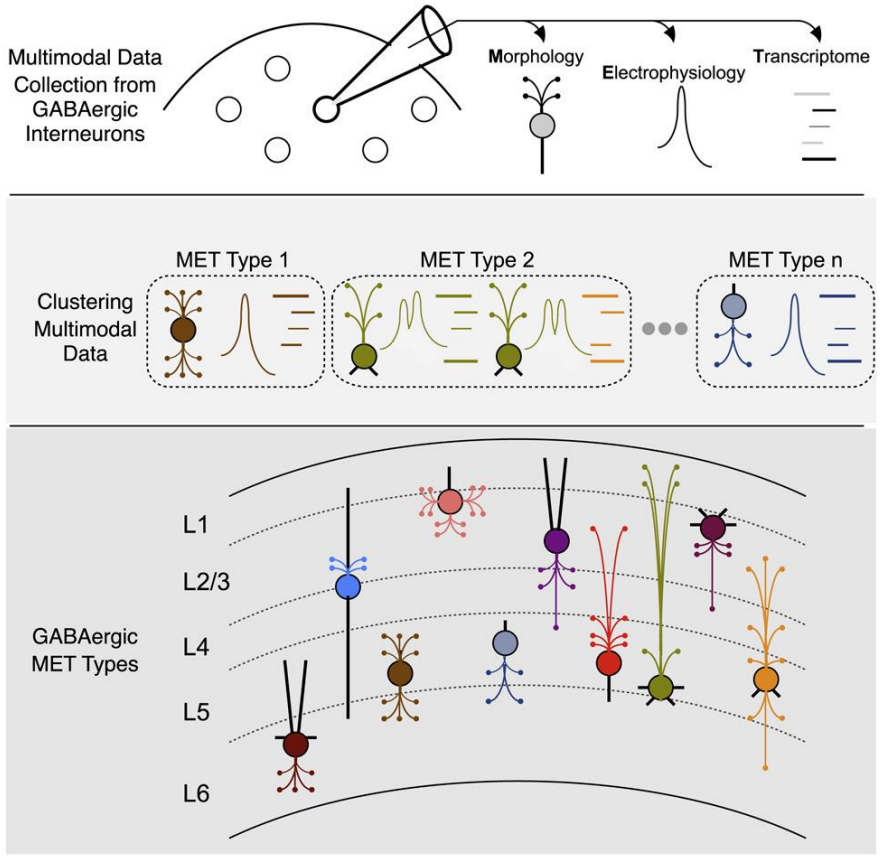
Mouse visual cortex
(4020 neurons)

- Gene expression
- Electrophysiology

Multi-modal feature

Classification

(brain layer and electrophysiological property)



Findings:

Neurons can be defined into **28** types

Distinguished by their layer-specific connection patterns

Manifold-Based Geometrical Techniques

Data source: multi-modal data from mouse brain neuronal cells

Article

Integrated Morphoelectric and Transcriptomic Classification of Cortical GABAergic Cells

Characterized types

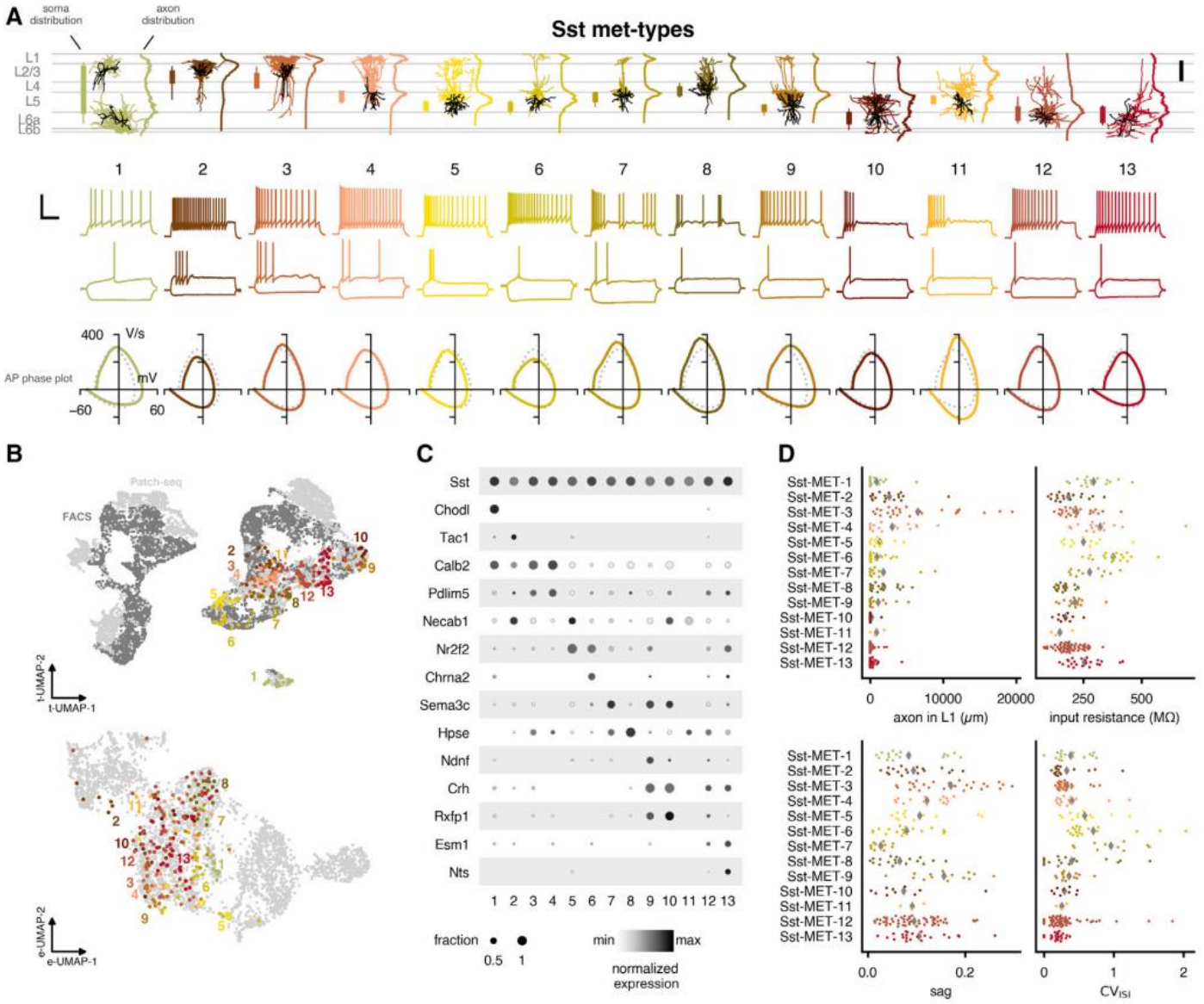
→ Congruent met-types

1) **electrophysiological** and

2) **transcriptomic** properties

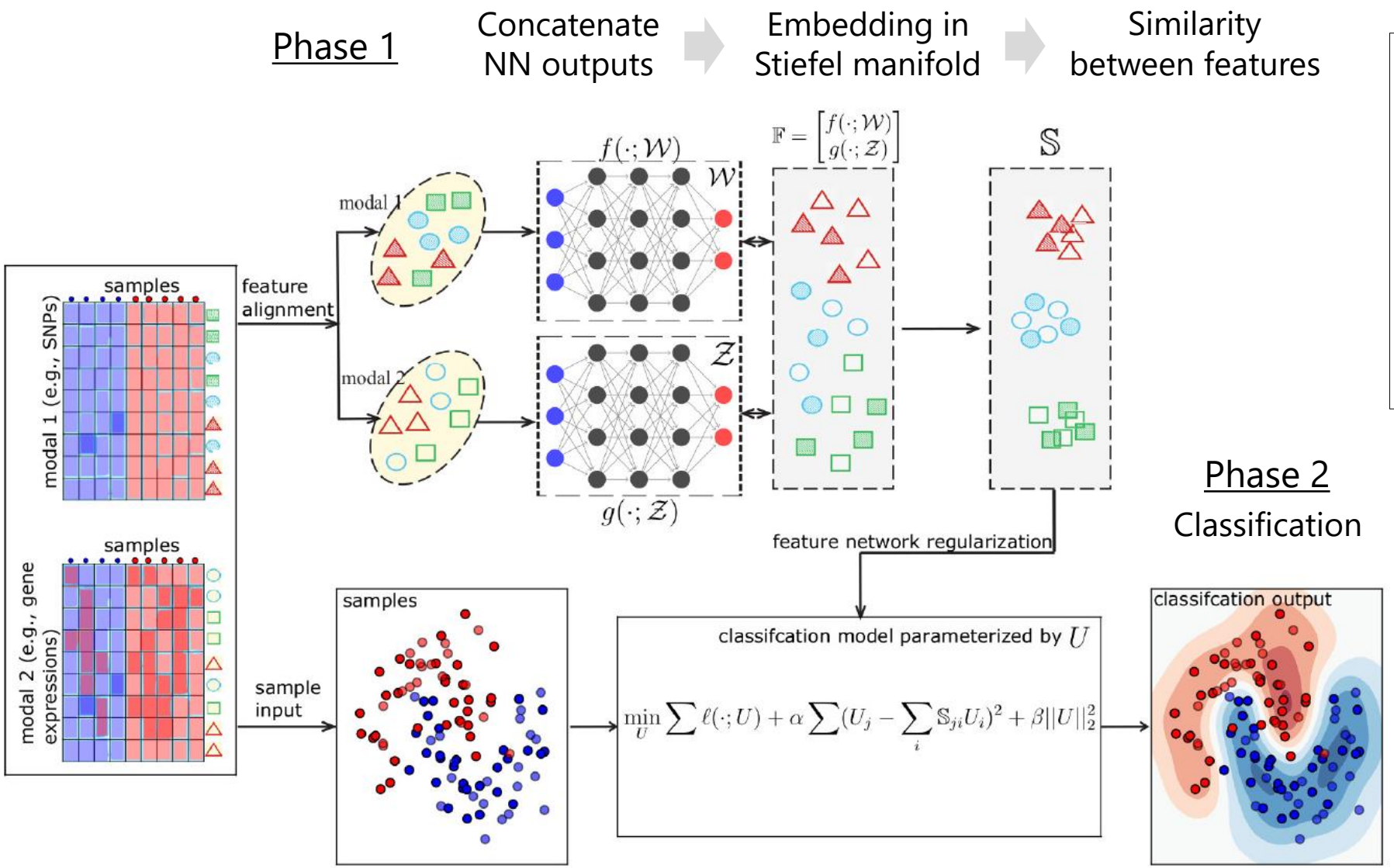
↓

Different met-types innervate different **cortical layers**



Manifold-Based Geometrical Techniques

Overview of deepManReg



1) Alignment of multi-modal features onto a **common latent space**

2) Embedding of data as points on **the Stiefel manifold** based on a set of orthonormal vectors

Manifold-Based Geometrical Techniques

deepManReg: Phase 1

Samples:

Single cells of neurons in mouse visual cortex

Sample **(3,654)**

Gene **(500)**

Modal 1:

Gene expression

$$X = \{x_i\}_{i=1}^{n}$$

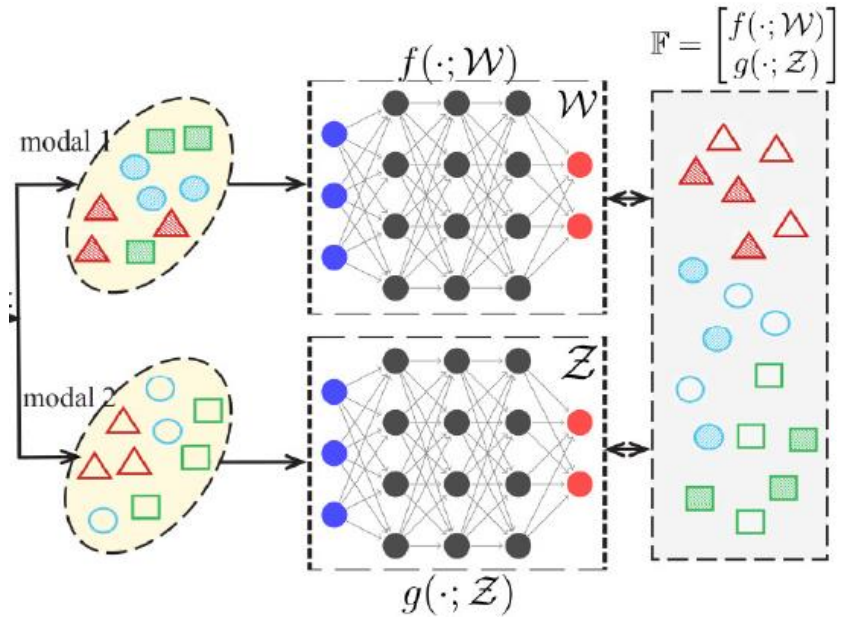
Sample **(3,654)**

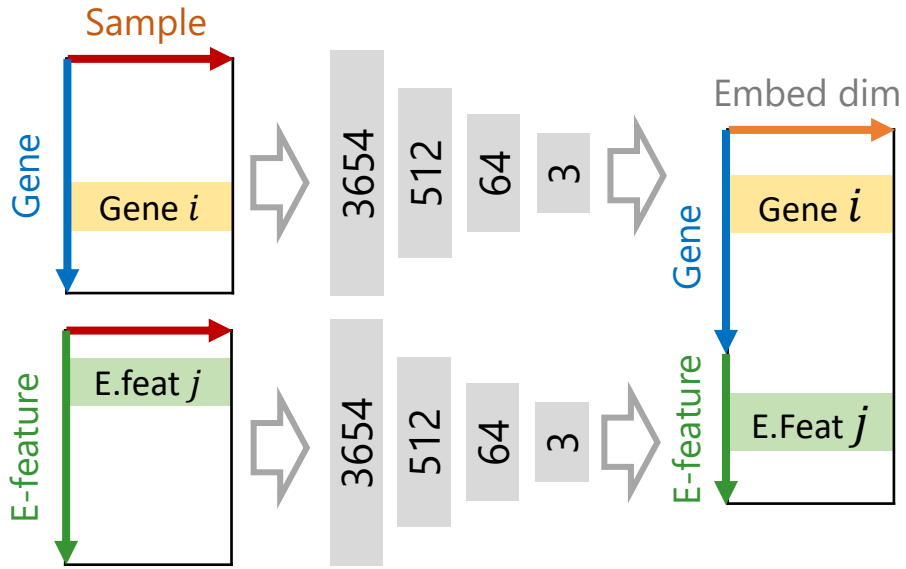
E-feature **(41)**

Modal 2:

Electrophysiological features

$$Y = \{y_j\}_{j=1}^{m}$$





1) Alignment of multi-modal features onto a **common latent space**

2) Embedding of data as points on **the Stiefel manifold** based on a set of orthonormal vectors

Laplacian eigenmaps

$$\min tr(\hat{\mathbb{F}}^T \hat{L} \hat{\mathbb{F}})$$

$$\text{s.t.} \hat{\mathbb{F}}^T \hat{\mathbb{F}} = \mathbb{I}$$

$$\hat{\mathbb{F}} = \mathbb{F} D^{1/2} \text{ and } \hat{L} = D^{-1/2} L D^{-1/2}.$$

Manifold-Based Geometrical Techniques

Phase 1: Manifold Alignment, Canonical Correlation Analysis (CCA)

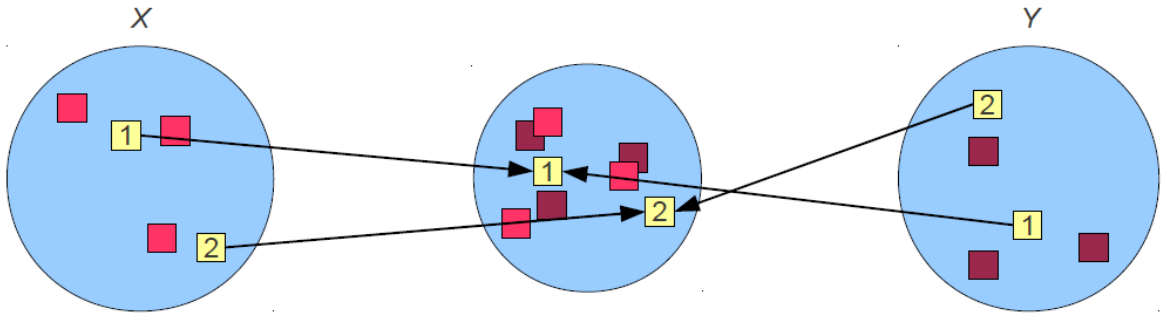


Figure 5.2: Given two datasets X and Y with two instances from both dataset that are known to be in correspondence, manifold alignment embeds all of the instances from each dataset in a new space where the corresponding instances are constrained to be equal (or at least close to each other) and the internal structures of each dataset are preserved.

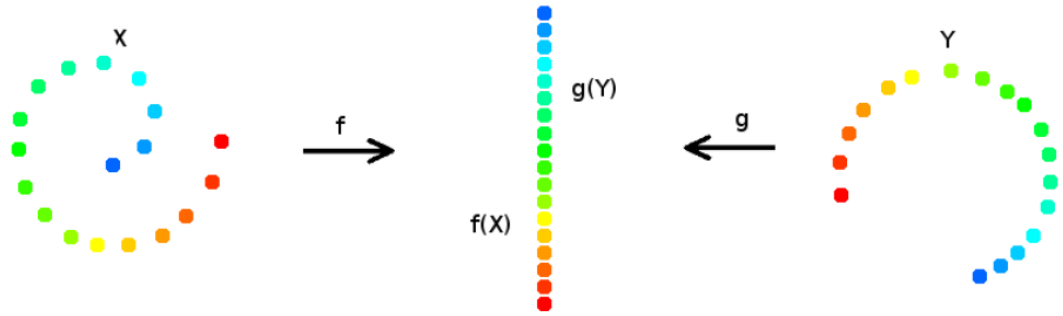
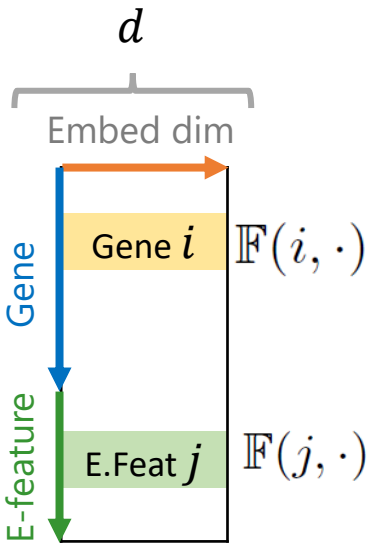


Figure 5.3: An illustration of the problem of manifold alignment. The two datasets X and Y are embedded into a single space where the corresponding instances are equal and local similarities within each dataset are preserved.

see a description written by Chang Wang, Peter Krat, and Sridhar Mahadevan. https://people.csail.mit.edu/pkrafft/papers/wang-et-al-2010-manifold.pdf

Manifold-Based Geometrical Techniques

Phase 1: Manifold Alignment, Canonical Correlation Analysis (CCA)



Cost function: Weighted Euclidean distance

$$\ell(\mathbb{F}) = \sum_{i,j}\sum_{k}\left(\mathbb{F}(i,k)-\mathbb{F}(j,k)\right)^2 W(i,j) \qquad W(i,j)=\begin{cases}1, & \text{if } Xi \text{ and } Yj \text{ are correspondence to each other}\\ 0, & \text{otherwise}\end{cases}$$

$$= tr(\mathbb{F}^T L\mathbb{F})$$

$\mathbf{D}$ is an $\sum_i n_i \times \sum_i n_i$ diagonal matrix with $\mathbf{D}(i,i) = \sum_j \mathbf{W}(i,j)$.

$\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the joint graph Laplacian

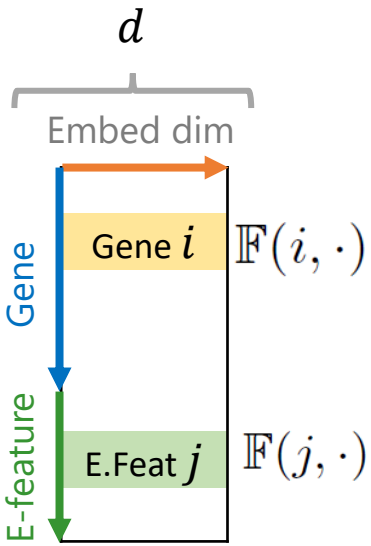
Laplacian eigenmaps

References

- Wang C, Krafft P and Mahadevanand S. **Manifold Alignment.** *Manifold Learning: Theory and Applications*, 2011. https://people.csail.mit.edu/pkrafft/papers/wang-et-al-2010-manifold.pdf
- Belkin Mikhail and Niyogi Partha. **Laplacian eigenmaps for dimensionality reduction and data representation**. *Neural Computation*, **15** (6):1373–1396, 2003.

Manifold-Based Geometrical Techniques

Phase 1: Manifold Alignment, Canonical Correlation Analysis (CCA)



Cost function: Weighted Euclidean distance

$$\ell(\mathbb{F}) = \sum_{i,j}\sum_{k}\left(\mathbb{F}(i,k)-\mathbb{F}(j,k)\right)^2 W(i,j) \qquad W(i,j)=\begin{cases}1, & \text{if } Xi \text{ and } Yj \text{ are correspondence to each other}\\ 0, & \text{otherwise}\end{cases}$$

$$= tr(\mathbb{F}^T L\mathbb{F})$$

$\mathbf{D}$ is an $\sum_i n_i \times \sum_i n_i$ diagonal matrix with $\mathbf{D}(i,i) = \sum_j \mathbf{W}(i,j)$.

$\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the joint graph Laplacian

This problem cannot be solved directly. Herein, the **orthogonal constraint** is imposed, thereby leading to avoiding the trivial solution ($\mathbb{F} = 0$). Thus, manifold alignment is possible to be performed by using **Laplacian eigenmaps** on the joint graph Laplacian (see a description written by Chang Wang, Peter Krat, and Sridhar Mahadevan, as shown later).

$$\min tr(\mathbb{F}^T L\mathbb{F})$$
$$\text{s.t.}\mathbb{F}^T D\mathbb{F} = \mathbb{I}$$

$\mathbb{I}$ is the $d \times d$ identity matrix.

$$\min tr(\hat{\mathbb{F}}^T \hat{L}\hat{\mathbb{F}})$$
$$\text{s.t.}\hat{\mathbb{F}}^T\hat{\mathbb{F}} = \mathbb{I}$$

$\hat{\mathbb{F}} = \mathbb{F}D^{1/2}$ and $\hat{L} = D^{-1/2}LD^{-1/2}$.

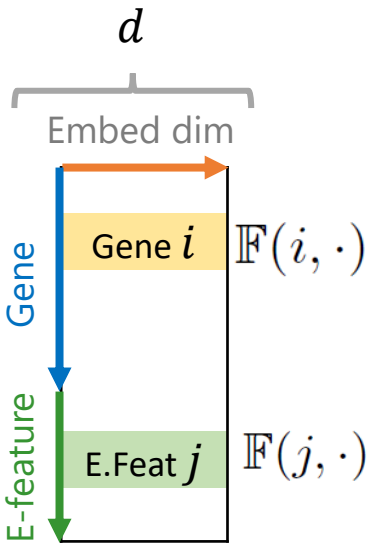
This can be solved by

Eigen Decomposition

Laplacian eigenmaps

Manifold-Based Geometrical Techniques

Phase 1: Manifold Alignment, Canonical Correlation Analysis (CCA)



これはちょうど Stiefel 多様体！
そこで Eigen Decomp. よりも、
Stiefel 多様体の上で
Gradient を計算して最適化しよう！

ch other

This problem cannot be solved directly. Herein, the **orthogonal constraint** is imposed, thereby leading to avoiding the trivial solution ($\mathbb{F} = 0$). Thus, manifold alignment is possible to be performed by using **Laplacian eigenmaps** on the joint graph Laplacian (see a description written by Chang Wang, Peter Krat, and Sridhar Mahadevan, as shown later).

$$\min tr(\mathbb{F}^T L \mathbb{F})$$
$$\text{s.t.} \mathbb{F}^T D \mathbb{F} = \mathbb{I}$$

$\mathbb{I}$ is the $d \times d$ identity matrix.

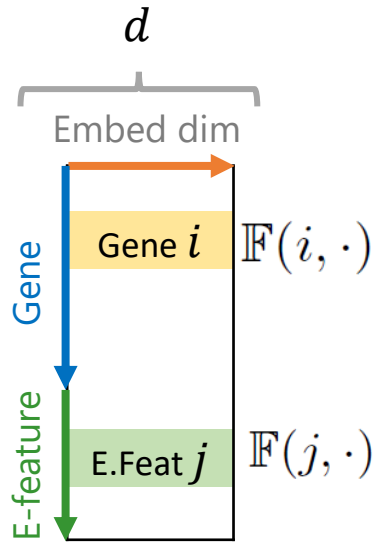
$$\min tr(\hat{\mathbb{F}}^T \hat{L} \hat{\mathbb{F}})$$
$$\text{s.t.} \hat{\mathbb{F}}^T \hat{\mathbb{F}} = \mathbb{I}$$

$\hat{\mathbb{F}} = \mathbb{F} D^{1/2}$ and $\hat{L} = D^{-1/2} L D^{-1/2}$.

This can be solved by
Eigen Decomposition

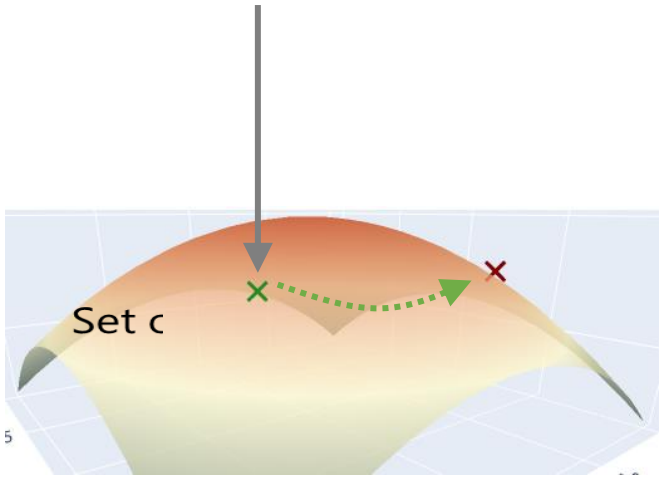
Manifold-Based Geometrical Techniques

Phase 1: optimization on the manifold



$\mathbb{F}(i,\cdot)$

$\mathbb{F}(j,\cdot)$



Cost function

$$\min tr(\hat{\mathbb{F}}^T \hat{L} \hat{\mathbb{F}})$$
$$\text{s.t.} \hat{\mathbb{F}}^T \hat{\mathbb{F}} = \mathbb{I}$$

Laplacian eigenmaps

Eigen Decomposition
(high computational cost)

↓

Optimization on the manifold

Manifold Alignment

Stiefel manifold $S_{n,p}$
(Stiefel E, *Commentarii Mathematici Helvetici*, 1935)

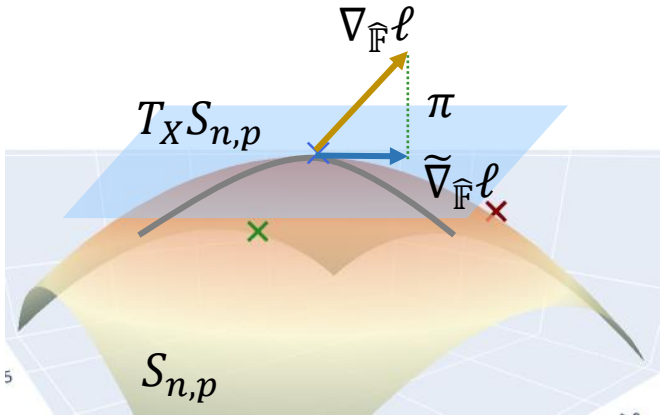
Set of $n \times p$ matrices satisfied $X^\top X = \mathbb{I}$

$$S_{n,p} := \{X \in \mathbb{R}^{n \times p} : X^T X = \mathbb{I}\}$$

Feature → single point on $S_{n,p}$

Manifold-Based Geometrical Techniques

Gradient Descent in Manifold



コスト関数 $\ell(\mathbb{F}) = tr(\hat{\mathbb{F}}^{\mathsf{T}}\hat{L}\hat{\mathbb{F}})$ は $\mathbb{R}^{n\times p}$ 上で定義された関数であり，$S_{n,p}$ 上で定義されたものではない

勾配 $\nabla\ell$ が**接空間 $T_X S_{n,p}$ 上にあるとは限らない**

→ 接空間上に正射影 (orthogonal projection)

$$\pi\colon\ \mathbb{R}^{n\times p} \to T_X S_{n,p}$$

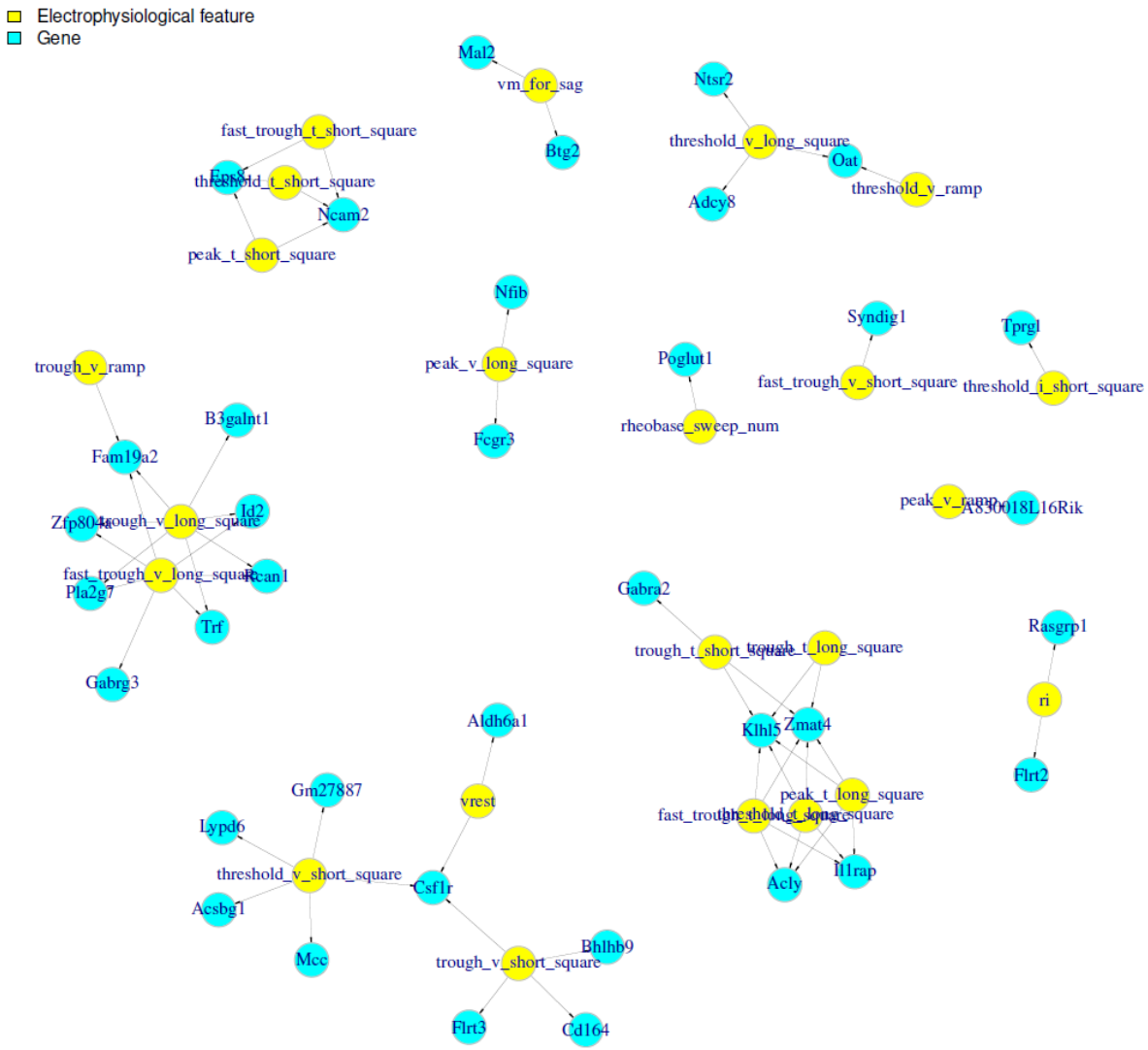
$$\tilde{\nabla}_{\hat{\mathbb{F}}}\ell = \pi(\nabla_{\hat{\mathbb{F}}}\ell) = \underset{X\in T_M(S_{m+n,d})}{\arg\min} \|\nabla_{\hat{\mathbb{F}}}\ell - X\|_F^2$$

$$= \hat{\mathbb{F}}skew(\hat{\mathbb{F}}^T\nabla_{\hat{\mathbb{F}}}\ell) + (I - \hat{\mathbb{F}}\hat{\mathbb{F}}^T)\nabla_{\hat{\mathbb{F}}}\ell.$$

$$\text{where } skew(\hat{\mathbb{F}}^T\nabla_{\hat{\mathbb{F}}}\ell) = \frac{1}{2}\left(\hat{\mathbb{F}}^T\nabla_{\hat{\mathbb{F}}}\ell - (\nabla_{\hat{\mathbb{F}}}\ell)^T\hat{\mathbb{F}}\right).$$

Manifold-Based Geometrical Techniques

Similarity between different modal data



Similarity between features

$$\mathbb{S} = \frac{1}{1+\mathbb{D}}$$

$\mathbb{D}$: distance matrix

$$\mathbb{D}_{ij} = \sqrt{\sum_k \left(\mathbb{F}(i,k) - \mathbb{F}(j,k)\right)^2}$$

Normalization

$$\sum_j \mathbb{S}_{ij} = 1$$

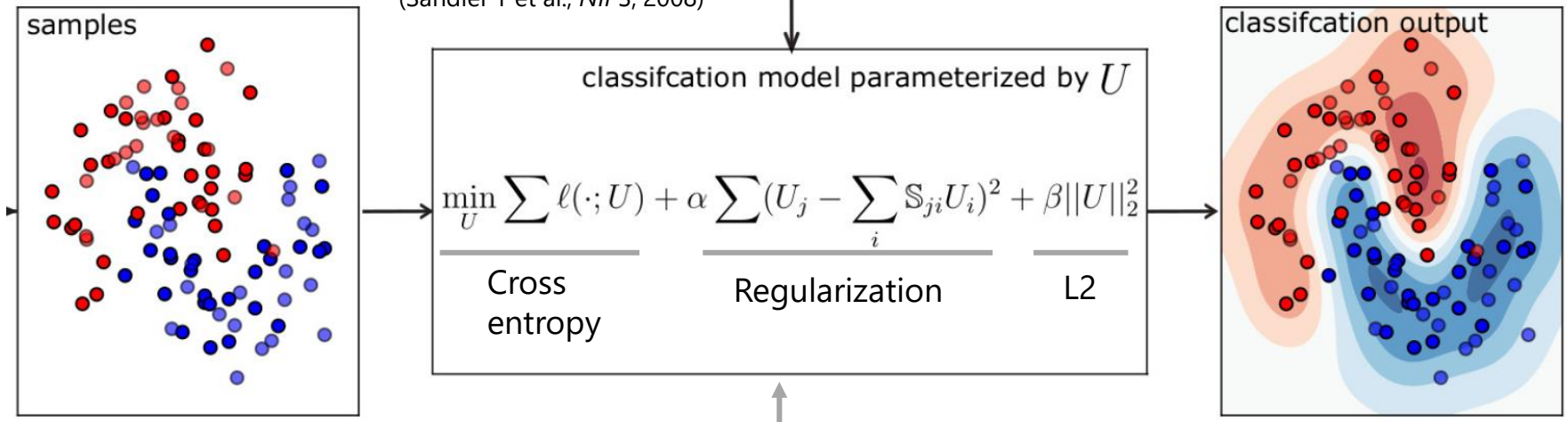
(Transition probability)

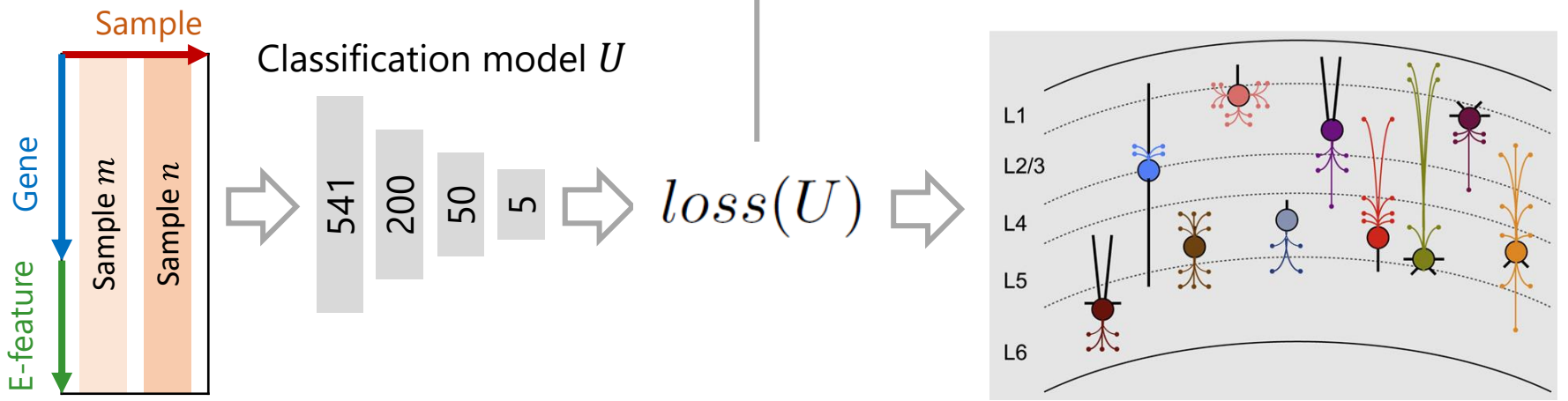
A network consists of different modal features

Manifold-Based Geometrical Techniques

Phase 2: classification

Regularized Learning with Networks of Features
(Sandler T et al., *NIPS*, 2008)





Prediction: layer in visual cortex

Manifold-Based Geometrical Techniques

deepManReg: summary

- Multi-Modal なデータから，**Features を抽出して、それらの類似性による Regularization** によってサンプルを分類

- 異なる Modals の Features を比較するため，それらの Features を同じ空間にマップした。

  Laplacian eigenmaps → Stiefel manifold (manifold alignment)

- 異なる Modal の Features をひとつの空間に Embed して比較する解析法は重要！　（理論に改善の余地もある）

Manifold-Based Geometrical Techniques

Reference

- Nguyen ND, Huang J and Wang D. **A Deep Manifold-regularized Learning Model for Improving Phenotype Prediction from Multi-modal Data.** *Nature Computational Science*, 2022.
- Wang C, Krafft P and Mahadevanand S. **Manifold Alignment.** *Manifold Learning: Theory and Applications*, 2011. https://people.csail.mit.edu/pkrafft/papers/wang-et-al-2010-manifold.pdf
- Belkin Mikhail and Niyogi Partha. **Laplacian eigenmaps for dimensionality reduction and data representation**. *Neural Computation*, **15** (6):1373–1396, 2003.
- Bonnabel S. **Stochastic gradient descent on Riemannian manifolds.** *IEEE Transactions on Automatic Control*, 2013.

- 杉田勝実, 岡本良夫, 関根松夫. **理論物理のための微分幾何学**. 森北出版, 2007.

Our Tactics

Various Properties and Features in Physics, Chemistry, and Biology

Deep Learning (DL) AI Systems and Machine Learning (ML) Techniques

Data Analysis on Curved, General Spaces (e.g. the Riemann Manifold)
→ Unification of Multimodal Data

Ex) Multi Omics Data, etc.

Multi-Scale Computer Simulations
→ Data Augmentation

Ex) Quantum Simulation, Non-Equilibrium Molecular Dynamics (MD) Simulation, etc.

Fundamentals and Principles in Natural Sciences

Fundamental Mathematical and Physical Sciences
Oriented to Joined AI and First Principles-Driving DD Works

Fundamental Lecture	主な内容		教科書
数学・基礎	複素数	Fourier Analysis	物理のための数学(岩波物理入門コース)
	多変数関数の微積分	Vector Analysis	物理の数学 (岩波基礎物理シリーズ)
	線形代数・基礎		1冊でマスター 大学の線形代数

	主な内容	教科書	参考書
物理学・基礎	力学・解析力学	物理学(サイエンス社) 基幹物理学(培風館)	よくわかる解析力学
	電磁気学		理論電磁気学
	熱力学		マッカーリ and サイモン 物理化学—分子論的アプローチ
	統計物理学		マッカーリ and サイモン 物理化学—分子論的アプローチ
	量子物理学		マッカーリ and サイモン 物理化学—分子論的アプローチ

場とは何か？
空間とは何か？

State Space Model ← 状態空間
Curved Space ← 曲がった空間
（Flat な空間の方が “まれ” なのかも！？）

PyQ +

MD and MI

TextBook

Molecular Dynamics	タンパク質計算科学 — 基礎と創薬への応用	コンピュータ・シミュレーションの基礎(第2版): 分子のミクロな性質を解明するために
Quantum Chemistry	新しい量子化学—電子構造の理論入門〈上〉	タンパク質密度汎関数法
Machine Learning	scikit-learn と TensorFlow による実践機械学習	ベイズ深層学習

資料

Manifold-Based Geometrical Techniques

Phase 1: Manifold Alignment, Canonical Correlation Analysis (CCA)

Ex) Instance Recognition
Similar Image Search
← Efficient methods employing a Hash Table

1) Measurement of the distances between a query and all data in a database.

2) Sort of the data in the database according to the distances measured.

Spectral Hashing :
Ref) Weiss, Y., Torralba, A., and Fergus, R., Spectral Hashing, In *NIPS* (2008).

1) Alignment of multi-modal features onto a **common latent space**

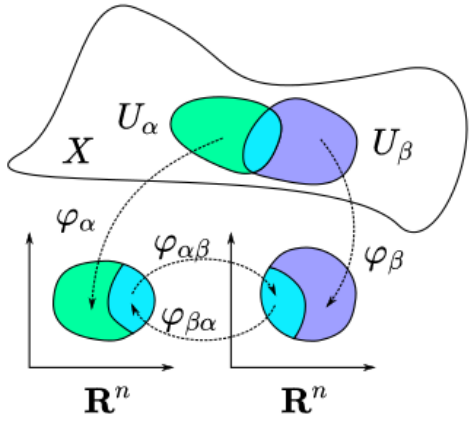
2) Embedding of data as points on **the Stiefel manifold** based on a set of orthonormal vectors

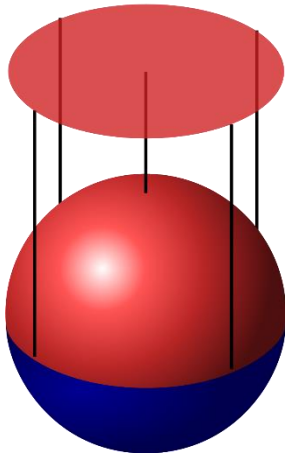
Laplacian eigenmaps

Ref) (found in this paper)
1) Hotelling Harold, "Relations between two sets of variates". In *Breakthroughs in statistics*, pages 162–190. Springer, 1992.
2) Nguyen Nam D and Wang Daifeng, "Multiview learning for understanding functional multiomics". *PLoS computational biology*, **16** (4):e1007677, 2020. [PubMed: 32240163]

Manifold-Based Geometrical Techniques

Manifold





https://en.wikipedia.org/wiki/Manifold
https://en.wikipedia.org/wiki/Differentiable_manifold

Topological space

A geometrical space which "closeness" (topology) can be defined for each point as a set of open sets (neighborhoods)

- Example of topology: **distance** (metric topology)

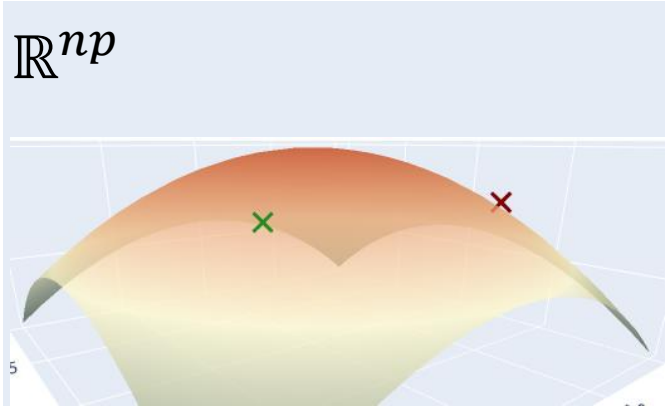
Manifold

A topological space in which a **topology-preserved** bijective map $\varphi: U \to \mathbb{R}^n$ from a neighborhood of each point U to **Euclidean space** $\mathbb{R}^n$

- Map φ is called **local coordinates**
- Example of manifold: 2-dimension sphere $\mathbb{S}^2$

Manifold-Based Geometrical Techniques

Embedding in Stiefel manifold



$S_{n,p} := \{X \in \mathbb{R}^{n\times p} : X^T X = \mathbb{I}\}$

$n \times p$ 行列 $\mathbb{R}^{n\times p}$ は適切に位相を定めることで，
$\mathbb{R}^{np}$ に埋め込める

Stiefel 多様体 $S_{n,p}$ は $\mathbb{R}^{np}$ 内の**部分空間**として
埋め込まれる

$$\dim(S_{n,p}) = np - \frac{p(p+1)}{2}$$

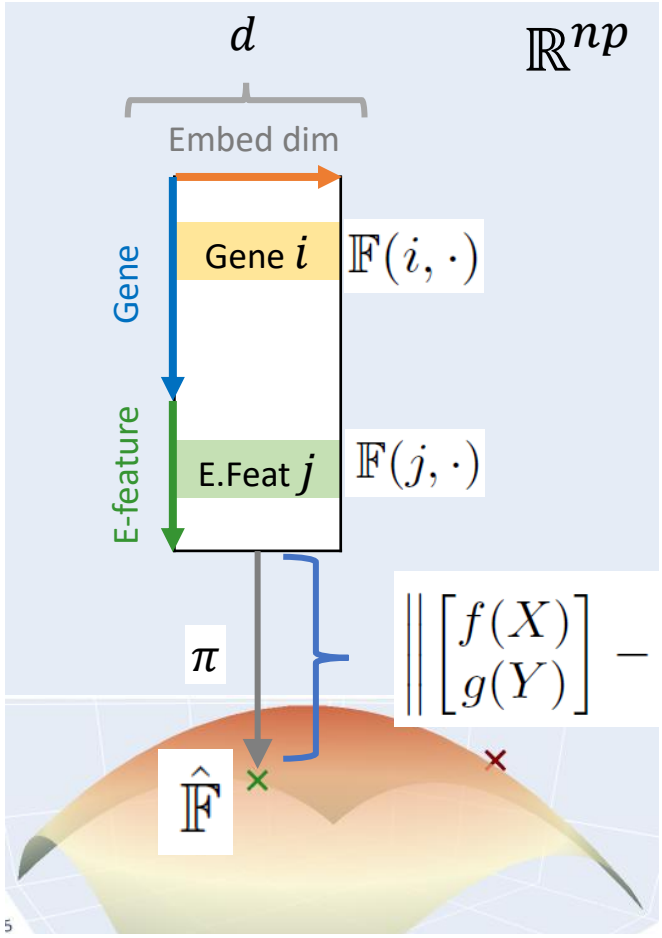
ここでは，$\mathbb{R}^{np}$ 及び $S_{n,p}$ の位相として Euclid 距離
（Frobenius norm）$\|\cdot\|_F$ による位相を用いた

$$X = \{X_{ij}\}, Y = \{Y_{ij}\} \in \mathbb{R}^{n\times p} \qquad d(X,Y) = \|X - Y\|_F = \sqrt{\sum_i^n \sum_j^p (X_{ij} - Y_{ij})^2}$$

An element in the **Stiefel manifold** $S_{n,p}$ is defined as (W; $\boldsymbol{e}_i$, … , $\boldsymbol{e}_p$), *i.e.*,
the p dimensional Real Vector Subspace W ($\subset \mathbb{R}^n$)
in the n dimensional Real Vector Space $\mathbb{R}^n$,
with the Orthonormal Basis Set { $\boldsymbol{e}_i$ } (i = 1, … p), $\boldsymbol{e}_i \in \mathbb{R}^n$.

Manifold-Based Geometrical Techniques

Projection to Stiefel manifold



$\mathbb{F}$ はDNNからの出力。
$\mathbb{R}^{np}$ 内にはあるが $S_{n,p}$ **内には無い**

前述の位相を用いて
$\mathbb{F}$ からの**距離が最短**である $S_{n,p}$ 内の点に射影（**正射影**）

$$\hat{\mathbb{F}} = \pi \circ \begin{bmatrix} f(X) \\ g(Y) \end{bmatrix} = \underset{X \in S_{m+n,d}}{\arg\min} \left\| \begin{bmatrix} f(X) \\ g(Y) \end{bmatrix} - X \right\|_F^2$$

Solution

$$\hat{\mathbb{F}} = U \mathbb{I}_{m+n,d} V^T$$

where $\begin{bmatrix} f(X) \\ g(Y) \end{bmatrix} = U \Sigma V^T$ is the SVD decompostion of $\begin{bmatrix} f(X) \\ g(Y) \end{bmatrix}$

$$S_{n,p} := \{X \in \mathbb{R}^{n \times p} : X^T X = \mathbb{I}\}$$

Manifold-Based Geometrical Techniques

deepManReg: algorithm

Algorithm 1: Deep Manifold Alignment

input : data for two modalities X & Y
params: training step T, learning rate η
output : parameters $\mathcal{W}_{T+1}$ & $\mathcal{Z}_{T+1}$

initialize $\mathcal{W}_0$ & $\mathcal{Z}_0$;
for $t = 0 : T$ **do**
  // forward pass
  $f_t \leftarrow f(X; \mathcal{W}_t)$;
  $g_t \leftarrow g(Y; \mathcal{Z}_t)$;
  $R_t \leftarrow \begin{bmatrix} f_t \\ g_t \end{bmatrix}$;
  Calculate L ; // L is the joint Laplacian
  // project the output onto Stiefel manifold
  $\hat{\mathbb{F}}_t \leftarrow U_t I V_t^T$ where $R_t = U_t \Sigma_t V_t^T$ is the SVD decomposition of R_t and $\mathbb{I}$ is the identity matrix;
  $\ell \leftarrow tr(\hat{\mathbb{F}}_t^T L \hat{\mathbb{F}}_t)$; // compute loss
  $e_t \leftarrow \nabla_{\hat{\mathbb{F}}_t} \ell$; // compute Euclidean gradient
  // project Euclidean gradient onto the tangent space of Stiefel manifold
  $p_t \leftarrow \hat{\mathbb{F}}_t skew(\hat{\mathbb{F}}_t^T e_t) + (I - \hat{\mathbb{F}}_t \hat{\mathbb{F}}_t^T) e_t$;
  // backpropagate the Riemannian gradient
  $\mathcal{W}_t, \mathcal{Z}_t \leftarrow \text{backprop}(p_t)$ $\mathcal{W}_{t+1} \leftarrow \mathcal{W}_t + g(\mathcal{W}_t, \eta, t)$;
  $\mathcal{Z}_{t+1} \leftarrow \mathcal{Z}_t + g(\mathcal{Z}_t, \eta, t)$ where g is an optimizer (e.g., SGD)
end

Input data X, Y

↓

Concatenate NN output $[f(X)\ g(Y)]$

↓

$S_{n,p}$ に正射影 $(\hat{\mathbb{F}})$

↓

コスト関数 ℓ を計算

↓

$\hat{\mathbb{F}}$ で偏微分 $(\nabla_{\hat{\mathbb{F}}} \ell)$

↓

接空間に正射影 p_t

↓

勾配降下法でf, g のパラメータを更新