Neural tangent kernel の HGM による計算

超幾何関数による計算

A. Sakoda, N. Takayama

arxiv: 2410.23626

T. §1. 予備知識

Neural net (NN) とは?

Kernel 法とは?

基本定理

$$\left\langle \frac{\partial f}{\partial \theta}(x), \frac{\partial f}{\partial \theta}(x') \right\rangle =: K(x,x')$$

$$\longrightarrow H(x,x')$$

S. §2. H の定義

dual activation

related works on dual activation

S. §3. HGM による dual activation の計算法. 計算実験

T. §4. Closed formula by H.G. ft.

§1. 予備知識

NN $f(\theta, x)$ とは?

幅

$$\mathbb{R}^{d_i} \longrightarrow \mathbb{R}^{d_{i+1}} \longrightarrow \mathbb{R}^{d_{i+1}}$$

affine

$$W^{(i)} x + b^{(i)}$$

bias

成分毎に

a. activation

つまり

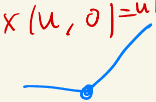
$$y \longmapsto \begin{pmatrix} a(y_1) \\ a(y_2) \\ \vdots \end{pmatrix}$$

これを合成したもの

を $f(\theta, x)$ 　 θ は W や b など.

ex. $a(u) = \max(u, 0) = uY(u)$

ReLU



学習とは? 補関:

入力データ $x^{(1)}, x^{(2)}, \cdots, x^{(N)}$

出力データ $y^{(1)}, y^{(2)}, \cdots, y^{(N)} \in \mathbb{R}$

から、パラメータ θ を求めること.

NN $f(\theta, x)$ とは？

幅

$$\mathbb{R}^{d_i} \longrightarrow \mathbb{R}^{d_{i+1}} \longrightarrow \mathbb{R}^{d_{i+1}}$$

affine

$$W^{(i)} x + b^{(i)}$$

成分毎に a. activation

つまり

$$y \longmapsto \begin{pmatrix} a(y_1) \\ a(y_2) \\ \vdots \end{pmatrix}$$

これを合成したもの。

を $f(\theta, x)$　　θ は W や b など

ex. $a(u) = \max(u, 0) = uY(u)$

ReLU

学習とは？

補関

入力データ　$x^{(1)}, x^{(2)}, \cdots, x^{(N)}$

出力データ　$y^{(1)}, y^{(2)}, \cdots, y^{(N)} \in \mathbb{R}$

から、パラメータ θ をきめること。

パラメータをきめる kernel 法とは

1950, 1990~

$\boxed{\frac{\partial f}{\partial \theta}}$ を kernel ft に。

$$K(x, x') = \left\langle \frac{\partial f}{\partial \theta}(\theta_0, x), \frac{\partial f}{\partial \theta}(\theta_0, x') \right\rangle$$

fixed

$f(0, x) = 0$.　以下略.

仮定1　$$f(\theta, x) = \theta \cdot \frac{\partial f}{\partial \theta}(\theta_0, x)$$

仮定2　$$\theta = \sum_{j=1}^{N} \alpha_j \frac{\partial f}{\partial \theta}(\theta_0, x^{(j)})$$

未知

$\alpha_j \in \mathbb{R}$

$$\Longrightarrow f(\theta, x) = \sum_{j=1}^{N} \alpha_j K(x^{(j)}, x)$$

loss ft

$$l(\alpha) = \frac{1}{2} \sum_{i=1}^{N} \left(\sum_{j=1}^{N} K_{ij} \alpha_j - y^{(i)} \right)^2$$

$$K_{ij} = K(x^{(i)}, x^{(j)})$$

$$\frac{\partial l}{\partial \alpha_k} = \sum_{i=1}^{N} K_{ik} \left(\sum_{j=1}^{N} K_{ij} \alpha_j - y^{(i)} \right)$$

$\boxed{K_{ij}} \boxed{\alpha_j} = \boxed{y^i}$ が十分条件

パラメータをきめるkernel法とは

1950, 1990~

$\boxed{\frac{\partial f}{\partial \theta}}$ を kernel ft に.

$$K(x,x') = \left\langle \frac{\partial f}{\partial \theta}(\theta_0, x), \frac{\partial f}{\partial \theta}(\theta_0, x') \right\rangle$$

fixed

$f(0, x) = 0.$ 以下略.

仮定1 $f(\theta, x) = \theta \cdot \frac{\partial f}{\partial \theta}(\theta_0, x)$

仮定2 $\theta = \sum_{j=1}^{N} \alpha_j \frac{\partial f}{\partial \theta}(\theta_0, x^{(j)})$

未知. $\alpha_j \in \mathbb{R}$.

$$\Longrightarrow f(\theta, x) = \sum_{j=1}^{N} \alpha_j K(x^{(j)}, x)$$

loss ft

$$l(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\left(\sum_{j=1}^{N} K_{ij}\alpha_j - y^{(i)}\right)^2$$

$$K_{ij} = K(x^{(i)}, x^{(j)})$$

$$\frac{\partial l}{\partial \alpha_k} = \sum_{i=1}^{N} K_{ik}\left(\sum_{j=1}^{N} K_{ij}\alpha_j - y^{(i)}\right)$$ $= 0$となるには

$\boxed{K_{ij}}\boxed{\alpha_j} = \boxed{y^{i}}$ が十分条件.

$\boxed{\alpha} = K^{-1} y$ となるので、

$$f(\theta, x) = \left(K(x^{(1)}, x), \cdots, K(x^{(N)}, x)\right) K^{-1} y$$

$$K(x,x') = \left(\frac{\partial f}{\partial \theta}(\theta_0, x), \frac{\partial f}{\partial \theta}(\theta_0, x') \right)$$

Neural tangent kernel (NTK)

Th (Jacot et al 2018, Arora et al 2019, ...)

arxiv: 1904.11955

幅 $\longrightarrow \infty$ なら.

$$K(x, x') \longrightarrow \Theta(x, x')$$

in probability

(θ_0-space: $N(0, E)$)

正規分布

Θ を用いると、入出力データを用いて $f(\theta, x)$ を書ける! Θ をどう計算?

定理: Jacot et al 2018[a]. NN の幅が $d_i \to \infty$ ならほとんどすべての θ について $K(x, x') \sim \Theta(x, x')$ となる.

[a] https://arxiv.org/abs/1806.07572

ネットの記事など [1]. Θ (NTK) の定義.

$$\Sigma^{(0)}(x, x') = x^T x' + \beta^2, \tag{3}$$

$$\Lambda^{(h)}(x, x') = \begin{pmatrix} \Sigma^{(h-1)}(x, x) & \Sigma^{(h-1)}(x, x') \\ \Sigma^{(h-1)}(x', x) & \Sigma^{(h-1)}(x', x') \end{pmatrix} \tag{4}$$

$$\Sigma^{(h)}(x, x') = c_\sigma E_{(u,v)\sim N(0,\Lambda^{(h)})}[\sigma(u)\sigma(v)] + \beta^2 \tag{5}$$

$$\dot{\Sigma}^{(h)}(x, x') = c_\sigma E_{(u,v)\sim N(0,\Lambda^{(h)})}[\dot{\sigma}(u)\dot{\sigma}(v)] \tag{6}$$

$$\Theta(x, x') = \Theta^{(L)}(x, x') = \sum_{h=1}^{L+1} \left(\Sigma^{(h-1)}(x, x') \prod_{h'=h}^{L+1} \dot{\Sigma}(x, x') \right) \tag{7}$$

[1] https://oumpy.github.io/blog/2020/04/neural_tangents.html

Dual activation

$E_{(u,v)\sim N(0,\Lambda^{(h)})}[\sigma(u)\sigma(v)]$ (dual activation of σ) は積分で書くと

$$\hat{E}[\sigma(u)\sigma(v)] = \int_{\mathrm{R}^2} \sigma(u)\sigma(v)\exp(x_{11}u^2 + 2x_{12}uv + x_{22}v^2)dudv \quad (8)$$

$$E_{(u,v)\sim N(0,\Lambda^{(h)})}[\sigma(u)\sigma(v)] = \hat{E}[\sigma(u)\sigma(v)]\frac{\sqrt{\det(x)}}{\pi}, \quad \Lambda^{(h)} = -\frac{1}{2}x^{-1}. \quad (9)$$

- ReLU (rectified linear unit)[2]: $\sigma(u) = uY(u)$, $Y(u)$ Heaviside 関数.
- Sigmoid: $\sigma(u) = \frac{1}{1+\exp(-u)}$
- GeLU (Gaussian error linear unit): $\sigma(u) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{u}{\sqrt{2}}\right)\right)$

dual activation が計算できれば NTK が計算できる

[2] https://en.wikipedia.org/wiki/Activation_function

Dual activation の方法はいろいろ研究されている.

計算

- ReLU については具体的に次の式が知られている. Λ を $\begin{pmatrix} c_1^2 & c_1 c_2 r \\ c_1 c_2 r & c_2^2 \end{pmatrix}$, $c_1, c_2 \geq 0, |r| \leq 1$ とすると,

$$E_{(u,v)\sim N(0,\Lambda)}[\sigma(u)\sigma(v)] = \frac{r(\pi - \arccos(r)) + \sqrt{1-r^2}}{2\pi} \cdot c_1 c_2 \quad (10)$$

$$E_{(u,v)\sim N(0,\Lambda^{(h)})}[\dot{\sigma}(u)\dot{\sigma}(v)] = \frac{\pi - \arccos(r)}{2\pi} \quad (11)$$

- Han et al arxiv:2209.04121 は一般的な計算法として Gauss-Hermite quadrature を用いる方法を与えているほか, 多項式の場合の dual activation を与えている.
- その他の activation に対する研究も Han et al の論文にまとめられている.

- activation が多項式係数の線形常微分方程式の解の時に dual activation を計算する一般的アルゴリズムを与えた。

Algorithm 1

(Holonomic Gradient Method, HGM)
Input: $\sigma_1(u)$ と $\sigma_2(u)$ を零化する線型 *ODE* ℓ_1 ,ℓ_2.
Output: $\hat{E}[\sigma_1(u)\sigma_2(v)]$ の値 (*NTK* のときは $\sigma_1 = \sigma_2$).

1. ℓ_1 と ℓ_2 が $\mathbf{C}\langle u, v, \partial_u, \partial_v\rangle$ で生成する左イデアルに小山-竹村 *Th2 [arxiv:1211.6822]* を適用して *Holnomic* イデアル $I_1 \in D_5 = \mathbf{C}\langle x_{11}, x_{12}, x_{22}, y_1, y_2, \partial_{11}, \partial_{12}, \partial_{22}, \partial_1, \partial_2\rangle$ を得る.
2. 制限アルゴリズム *[Oaku 1997]* を用いて $I_2 := (I_1 + y_1 D_5 + y_2 D_5) \cap \mathbf{C}\langle x_{11}, x_{12}, x_{22}, \partial_{11}, \partial_{12}, \partial_{22}\rangle$ の生成元を見つける.
3. I_2 を *Pfaffian* 系に書く.
4. 初期値として F の $x_{11} = -1, x_{12} = 0, x_{22} = -1$ またはその周りの点での値を級数展開などを使って評価する.
5. *Pfaffian* 系を数値的に解く

$$\int_{\mathrm{R}^2} uY(u)vY(v)\exp(x_{11}u^2 + 2x_{12}uv + x_{22}v^2 + y_1u + y_2v)dudv$$

- 小山-竹村 2012 はこの積分の微分方程式を求める方法を与えている.
- $\hat{E}[\sigma_1(u)\sigma_2(v)]$ の $(x_{11}, x_{12}, x_{22}) = (-1, 0, -1)$ 周りでの級数展開は $\sum_{k\in\mathrm{N}_0^3} c_k x^k$, $x^k = (x_{11}+1)^{k_{11}} x_{12}^{k_{12}} (x_{22}+1)^{k_{22}}$ として

$$\begin{aligned} c_k &= \frac{2^{k_{12}}}{k_{11}!k_{12}!k_{22}!} \\ &\times \int_{-\infty}^{\infty} u^{2k_{11}+k_{12}}\sigma_1(u)\exp(-u^2)du \\ &\times \int_{-\infty}^{\infty} v^{2k_{22}+k_{12}}\sigma_2(v)\exp(-v^2)dv. \end{aligned} \tag{12}$$

これを使って初期値を評価する.

HGM はパラメータ付き積分の値を計算

- $(u\partial_u - 1) \bullet uY(u) = 0.$
- 小山-竹村 2012[3] より

$$\int_{R^2} uY(u)vY(v)\exp(x_{11}u^2 + 2x_{12}uv + x_{22}v^2 + y_1u + y_2v)dudv$$

は

$$\partial_1(-y_1 - 2(x_{11}\partial_1 + x_{12}\partial_2)) - 1, \quad (13)$$

$$\partial_2(-y_2 - 2(x_{12}\partial_1 + x_{22}\partial_2)) - 1, \quad (14)$$

$$\partial_{12} - 2\partial_1\partial_2, \quad (15)$$

$$\partial_{11} - \partial_1^2, \quad (16)$$

$$\partial_{22} - \partial_2^2, \quad (17)$$

を満たす ($\partial_{ij} = \partial/\partial x_{ij}$, $\partial_i = \partial/\partial y_i$). 生成する左イデアルを I_1.

[3] https://arxiv.org/abs/1211.6822

- 積分を $y_1 = y_2 = 0$ に制限した関数の満たす方程式は

$$I_2 = (I_1 + y_1 D + y_2 D) \cap \mathbf{C}\langle x_{11}, x_{12}, x_{22}, \partial_{11}, \partial_{12}, \partial_{22} \rangle$$

ここで $D = \mathbf{C}\langle y_1, y_2, x_{11}, x_{12}, x_{22}, \partial_1, \partial_2, \partial_{11}, \partial_{12}, \partial_{22} \rangle$.

- 素手の制限計算, たとえば

$$\begin{aligned} & -y_1\partial_1 - 1 - 2(x_{11}\partial_1^2 + x_{12}\partial_1\partial_2) - 1 \\ \to \quad & -y_1\partial_1 - 2(x_{11}\partial_{11} + (1/2)x_{12}\partial_{12}) - 1 - 1, \qquad \text{by (12) and (13).} \end{aligned}$$

- $g(x) = \hat{E}[\sigma(u)\sigma(v)](x)$, $\sigma = \mathrm{ReLU}$ が満たすのは

$$\begin{aligned} & 2x_{11}\partial_{11} + x_{12}\partial_{12} + 2, \\ & x_{12}\partial_{12} + 2x_{22}\partial_{22} + 2, \\ & 4\partial_{11}\partial_{22} - \partial_{12}^2 \end{aligned}$$

$$F = (1, \partial_{12})^T \bullet g.$$

$$\partial_{x_{11}} \bullet F - P_{11} F = 0, \partial_{x_{12}} \bullet F - P_{12} F = 0, \partial_{x_{22}} \bullet F - P_{22} F = 0$$

$$\begin{aligned}
P_{11} &= \begin{pmatrix} \frac{-1}{x_{11}} & \frac{-\frac{1}{2} x_{12}}{x_{11}} \\ \frac{2x_{12}}{x_{11}(x_{12}^2 - x_{22}x_{11})} & \frac{\frac{1}{2}(2x_{12}^2 + 3x_{22}x_{11})}{x_{11}(x_{12}^2 - x_{22}x_{11})} \end{pmatrix}, \\
P_{12} &= \begin{pmatrix} 0 & 1 \\ \frac{-4}{x_{12}^2 - x_{22}x_{11}} & \frac{-5x_{12}}{(x_{12}^2 - x_{22}x_{11})} \end{pmatrix}, \\
P_{22} &= \begin{pmatrix} \frac{-1}{x_{22}} & \frac{-\frac{1}{2} x_{12}}{x_{22}} \\ \frac{2x_{12}}{x_{22}(x_{12}^2 - x_{22}x_{11})} & \frac{\frac{1}{2}(2x_{12}^2 + 3x_{22}x_{11})}{x_{22}(x_{12}^2 - x_{22}x_{11})} \end{pmatrix}.
\end{aligned}$$

$F(-1, 0, -1) = (1/4, \pi/8)^T$. `scipy`, `solve_ivp`.

```
import("nk_restriction.rr");;
V=[y1,y2,x11,x12,x22]; DV=poly_dvar(V);
P1=poly_dmul(dy1,-y1-2*x11*dy1-2*x12*dy2,V)-1;
P2=poly_dmul(dy2,-y2-2*x12*dy1-2*x22*dy2,V)-1;
I=[P1,P2,dx11-dy1^2,dx22-dy2^2,dx12-2*dy1*dy2];
dp_gr_print(1);
Iprime=nk_restriction.restriction_ideal(I,V,DV,[1,1,0,0,0])

import("yang.rr");;
VV=[x11,x12,x22]; DVV=poly_dvar(VV);
yang.define_ring(["partial",VV]);
RII=map(dp_ptod,Iprime,DVV);
yang.verbose();
RG=yang.buchberger(RII);;
Std=[1,dx12];
Pf=yang.pfaffian(map(dp_ptod,Std,DVV),RG);
```

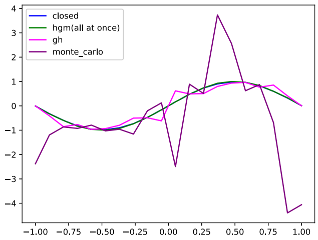
ReLU $\sigma = uY(u)$

Rank 2 system. 0.196s (Risa/Asir, AMD EPYC 7552 48-Core, 1.5GHz)

Method	Training time	Pred time
closed	1.500e-2	1.98e-2
Gauss-Herm	3.352e0	3.306e0
hgm	8.571e0	1.017e1
Monte-Carlo	8.597e1	1.149e2

	Kernel error
hgm	2.779e-8
Gauss-Herm	1.034e-3
Monte-Carlo	1.103e-3

	Pred error
hgm	2.815e-3
Gauss-Herm	4.164e-2
Monte-Carlo	4.039e-1



MNIST に含まれる画像の例

MNIST

MNIST 手書き文字データの 0 と 1 の画像に対して計算した結果. 訓練データは 100 件, テストデータは 20 件 Intel(R) Xeon(R) Gold 6426Y (800MHz) and NVIDIA A800 40GB.

Method	Eval time	Kernel error
closed	2.702e0	
Gauss-Herm	1.280e2	6.596e-2
hgm	1.727e2	1.456e-6
Monte-Carlo	2.904e3	1.814e-2

hgm-ntk の実装

- Θ の計算と Pfaffian の数値解析に工夫が必要
- 与えられた N 個のデータに対して $(x_i, x_j), i, j = 1, \ldots, N$ の内積を計算して記憶する
- 計算した内積から適切なものを選んで来て共分散行列を作りました記憶しておく
- 共分散行列が退化しているか, それに近い状態のときは HGM を使えないので除く (数値積分で期待値を計算する)
- 残った共分散行列を HGM の入力とするが, さらに 3 次元の点として見たときの距離が近すぎると ODE Solver が上手く動かないのでこれも除く (十分近い点の値をそのまま期待値として採用する)

- カーネル法はデータの件数 N に対して N^2 回の計算が必要なので高速化の工夫が必要
- 例えばデータの次元を M,X をデータを並べた $N \times M$ 行列とするとデータの内積を取る部分は行列で

$$XX^T$$

とかける.

- これは GPU によってかなり高速化ができるので MNIST の実験では GPU を使っている.
- MNIST の実験では他に HGM の入力を作る部分も GPU を使っている

まとめ

- Holonomic 関数であれば計算代数のアルゴリズムを使って NTK の近似計算ができる.
- 滑らかでないが Holonomic な活性化関数に対して有効な手法になり得る.

§4. closed formula.

<u>問題</u> dual activation を特殊関数であらわせ. Ⓗ の高速高精度計算.

Han et al 2022. arxiv:2209.04121

Activation	$\sigma(t)$	Reference for the NNGP	Reference for the NTK
Rectified monomials	$t^q \cdot \mathbb{1}_{\{t \geq 0\}}$	[44]	[44]
Error function	$\mathrm{erf}(t)$	[43]	[5]
ABReLU (Leaky ReLU)	$-A\min(t,0) + B\max(t,0)$	[50, 51, 42]	[50, 51, 42]
Exponential	$\exp(At)$	[52, 46]	[52, 46]
Hermite polynomials	$h_q(t)$	[46]	This work
Sinusoidal	$\sin(At+B)$	[45, 47, 53]	This work
Gaussian	$\exp\left(-At^2\right)$	[43]	This work
GeLU	$\frac{t}{2}\left(1+\mathrm{erf}\left(\frac{t}{\sqrt{2}}\right)\right)$	[48]	This work
ELU	$\mathrm{step}(t)t + \mathrm{step}(-t)\left(e^t - 1\right)$	[48]	This work
Normalized Gaussian	Unknown	[54]	This work
RBF	$\sqrt{2}\sin(\sqrt{2A}t + \frac{\pi}{4})$	[45]	This work
Gabor	$\exp(-t^2)\sin(t)$	This work	This work
Monomial	t^q	This work	This work
Polynomial	$\sum_{j=0}^q a_j t^j$	This work	This work

↳ $a(t) = \left(\sum_{j=0}^{q} a_j t^j\right) Y(t)$. ? our work.

$$\hat{E}[a_1(u)a_2(v)] = \int_{\mathbb{R}^2} a_1(u)a_2(v)\exp(x_{11}u^2 + 2x_{12}uv + x_{22}v^2)\,du\,dv$$

$a = a_1 = a_2$. a の dual activation

Th.
$$\hat{E}[u^m v^n Y(u)Y(v)] = (-x_{11})^{\alpha}(-x_{22})^{\beta} \times \Big[\frac{1}{4}\Gamma(\alpha)\Gamma(\beta)\,{}_2F_1\left(\alpha,\beta,\frac{1}{2};z\right) + \frac{1}{2}\Gamma\left(\alpha+\frac{1}{2}\right)\Gamma\left(\beta+\frac{1}{2}\right)\sqrt{z}\,\mathrm{sign}(x_{12})\cdot{}_2F_1\left(\alpha+\frac{1}{2},\beta+\frac{1}{2},\frac{3}{2};z\right)\Big]$$

$$\alpha = \frac{m+1}{2},\quad \beta = \frac{n+1}{2},\quad z = \frac{x_{12}^2}{x_{11}x_{22}}$$

$x_{11} < 0,\ x_{22} < 0 \qquad 0 \leq z < 1.$

$$\hat{E}[a_1(u)\,a_2(v)] = \int_{\mathbb{R}^2} a_1(u)a_2(v)\exp(x_{11}u^2 + 2x_{12}uv + x_{22}v^2)\,du\,dv$$

$a = a_1 = a_2$, a の dual activation

Th. $$\hat{E}[u^m v^n Y(u) Y(v)] = (-x_{11})^{\alpha}(-x_{22})^{\beta} \times$$

$$\Big[\frac{1}{4}\Gamma(\alpha)\Gamma(\beta)\,{}_2F_1(\alpha,\beta,\tfrac{1}{2};z)$$

$$+\frac{1}{2}\Gamma(\alpha+\tfrac{1}{2})\Gamma(\beta+\tfrac{1}{2})\sqrt{z}\,\mathrm{sign}(x_{12})$$

$$\cdot\,{}_2F_1(\alpha+\tfrac{1}{2},\beta+\tfrac{1}{2},\tfrac{3}{2};z)\Big]$$

$$\alpha = \frac{m+1}{2},\quad \beta = \frac{n+1}{2},\quad z = \frac{x_{12}^2}{x_{11}x_{22}}$$

$x_{11} < 0$, $x_{22} < 0$ $\quad 0 \le z < 1$.

Proof sketch.

① 小山-竹村. arxiv 1211.6822

より. $\hat{E}[u^m v^n Y(u) Y(v)]$ のみたす

holonomic 系は

$$\begin{cases} \begin{pmatrix} 2 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_{11}\partial_{11} \\ x_{12}\partial_{12} \\ x_{22}\partial_{22} \end{pmatrix} + \begin{pmatrix} m+1 \\ n+1 \end{pmatrix} \\ 4\partial_{11}\partial_{22} - \partial_{12}^2 \end{cases}$$

② これは. GKZ系 (A-HG).

series sol は、${}_2F_1$

③ $x_{12} = 0$ とすると、$\hat{E}$ は一重積分の積.

その値で係数をきめる。 //

Proof sketch.

① 小山-竹村. arxiv 1211.6822

より $\hat{E}[u^m v^n Y(u) Y(v)]$ のみたす

holonomic 系は

$$\begin{cases} \begin{pmatrix} 2 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_{11}\partial_{11} \\ x_{12}\partial_{12} \\ x_{22}\partial_{22} \end{pmatrix} + \begin{pmatrix} m+1 \\ n+1 \end{pmatrix} \\ 4\partial_{11}\partial_{22} - \partial_{12}^2 \end{cases}$$

② これは GKZ系 (A-HG).

series sol は ${}_2F_1$

③ $x_{12}=0$ とすると、$\hat{E}$ は一重積分の積.

その値で係数をきめる。 //

展望.

たとえば.

このTh を用いると. $a(u) = \sin u \, Y(u)$

の dual activation は 多変数超幾何.

超幾何の高速高精度数値計算

⇒ (deep) NTK の計算

$Y(u)\sin u$ の dual activation は $z = \frac{x_{12}^2}{x_{11}x_{22}}$ として

$$\frac{1}{4}(x_{11}x_{22})^{-1}\sum_{k=0}^{\infty}\frac{(1)_k}{(1/2)_k}z^k\,{}_1F_1\left(k+1,\frac{3}{2};\frac{1}{4x_{11}}\right){}_1F_1\left(k+1,\frac{3}{2};\frac{1}{4x_{22}}\right)$$

$$+\frac{\pi}{8}(x_{11}x_{22})^{-1}\mathrm{sign}(x_{12})\sqrt{z}\sum_{k=0}^{\infty}\frac{(3/2)_k}{(1)_k}z^k\,{}_1F_1\left(k+\frac{3}{2},\frac{3}{2};\frac{1}{4x_{11}}\right){}_1F_1\left(k+\frac{3}{2},\frac{3}{2};\frac{1}{4x_{22}}\right)$$