An Application of the Holonomic Gradient Method to the Neural Tangent Kernel

Akihiro Sakoda and Nobuki Takayama

2024.10.31

1 Introduction

A.Jacot et al [11] introduced a function $\Theta(x, x')$ that converges to the neural tanget kernel (NTK). Here, x, x' are data vectors. In order to construct this function, we need to evalute the expectations

$$E_{(u,v)\sim N(0,\Lambda^{(h)})}[\sigma(u)\sigma(v)] \tag{1}$$

$$E_{(u,v)\sim N(0,\Lambda^{(h)})}[\dot{\sigma}(u)\dot{\sigma}(v)]. \tag{2}$$

Here σ is an activator function of the neural network and $\Lambda^{(h)}$ is a 2×2 covariance matrix inductively defined. See, e.g., [1, (7)]. Each of these expectations is called a *dual activation* of σ and its derivative $\dot{\sigma}$ respectively. Note that these expectations can be expressed as definite integrals with parameters.

Attempts have been made to calculate these expectations for various activator functions, and closed forms have been found for many activator functions. Han et al [8] gives several new closed forms as well as a survey on the works on closed forms.

A system of linear partial differential equations of n variables is called a *holonomic system* when the dimension of its characteristic variety (the variety defined by the ideal generated by principal symbols) is n. A distribution is called a *holonomic distribution* if it is a solution of a holonomic system. In this paper, we note that when the activator function is a holonomic distribution, these expectations satisfy holonomic systems of linear partial differential equations and further show that these holonomic systems can be derived automatically by computer algebraic algorithms. We give the following new results based on this fact.

1. We give a method to evaluate these expectations using a numerical method for solving linear ordinary differential equations. This will provides a general method to calculate Θ when a new holonomic activator distribution is proposed.

2. When the activator distribution is a polynomial times a Heaviside function, this expectation can be expressed as a closed form in terms of the Gauss hypergeometric function.

3. Han et al [8] gives a general expression of the dual activation for a polynomial activator function. Smooth activators have Hermite polynomial expansions. They utilize this fact to give an approximate dual activation. We present a computer algebra method, which is well-known among computer algebra experts, to derive Hermite expansions.

The method of deriving a holonomic system and numerically evaluating definite integrals with parameters by its numerical analysis is called the holonomic gradient method (HGM) and has been applied to a variety of problems [26]. We refer to the book [9, chap 6] and papers [20], [21] as introductory documents. Although methods proposed in this paper falls into the HGM, our methods are specialized for the evaluation of (1) and (2) to make it more efficient, which is done by some improvements of numerical solvers for the HGM, by utilizing the result by Koyama and Takmura [12], and by restriction algorithms in computer algebra to derive holonomic systems of these expectations.

Related works: refer to [8] on a comprehensive survey on dual activation.

2 Computation of Θ

Jacot et al [11, Th 1] introduced a function Θ that approximates the neural tangent kernel. Arora et al [1, Th 3] gave a precise error analysis of the approximation. Following these papers, we briefly summarize the procedure to construct the function Θ.

Let $f(x, \theta)$ be a neural network whose input is x and parameter vector is θ. The neural tangent kernel (NTK) is a kernel function defined by

$$K(x, x') = \left\langle \frac{\partial f(x,\theta)}{\partial \theta}, \frac{\partial f(x',\theta)}{\partial \theta} \right\rangle \tag{3}$$

where $\frac{\partial f(x,\theta)}{\partial \theta}$ is the gradient vector and $\langle\ ,\ \rangle$ is the standard inner product.

The neural network f is a composition of linear functions and activator functions defined as follows. Let $x \in \mathbf{R}^d$ be an input and put $g^{(0)}(x) = x$, $d_0 = d$. Our fully connected neural network of L layers is inductively defined as follows

$$f^{(h)}(x) = W^{(h)} \cdot g^{(h-1)} \in \mathbf{R}^{d_h},\ g^{(h)} = \sqrt{\frac{c_\sigma}{d_h}} \sigma\left(f^{(h)}(x)\right) \in \mathbf{R}^{d_h},\ h = 1, 2, \ldots, L$$

Here, $W^{(h)} \in \mathbf{R}^{d_h \times d_{h-1}}$ is a weight matrix of the h-th layer, σ is an activator function, $c_\sigma = \left(E_{z\sim N(0,1)[\sigma(z)^2]}\right)^{-1}$ is the inverse of the expectation of σ^2 under the normal distribution with the mean 0 and the covariance 1. $\sigma((y_1, \ldots, y_{d_h})^T)$ means $(\sigma(y_1), \ldots, \sigma(y_{d_h}))^T)$. The output of the last layer is defined as

$$f(x, \theta) = f^{(L+1)}(x) = W^{(L+1)} \cdot g^{(L)}(x), \quad W^{(L+1)} \in \mathbf{R}^{1 \times d_L}.$$

Let us introduce the function Θ. We inductively define covariance matrices $\Lambda^{(h)}(x, x')$ as follows.

$$\Sigma^{(0)}(x, x') = x^T x', \tag{4}$$

$$\Lambda^{(h)}(x, x') = \begin{pmatrix} \Sigma^{(h-1)}(x, x) & \Sigma^{(h-1)}(x, x') \\ \Sigma^{(h-1)}(x', x) & \Sigma^{(h-1)}(x', x') \end{pmatrix} \tag{5}$$

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

$$\dot{\Sigma}^{(h)}(x, x') = c_\sigma E_{(u,v)\sim N(0,\Lambda^{(h)})}[\dot{\sigma}(u)\dot{\sigma}(v)] \tag{7}$$

Here, $\dot{\sigma}$ is the derivative of the activator function σ. The function $\Theta(x, x')$ approximating the neural tangent kernel is defined as

$$\Theta(x, x') = \Theta^{(L)}(x, x') = \sum_{h=1}^{L+1} \left(\Sigma^{(h-1)}(x, x') \prod_{h'=h}^{L+1} \dot{\Sigma}(x, x') \right) \tag{8}$$

Here, we put $\dot{\Sigma}^{(L+1)}(x, x') = 1$.

Assume that all elements of parameter θ are independent and identically distributed as $N(0, 1)$. When the width of the neural network is infinite $d_1, d_2, \ldots, d_L \to \infty$, the following theorems hold.

Theorem 1. [1, Th 3.1] *Fix $\epsilon > 0$ and $\delta \in (0, 1)$. Suppose $\sigma(z) = \max(0, z)$ and $\min_{h\in[L]} d_h \geq \Omega(\frac{L^6}{\epsilon^4} \log(L/\delta))$. Then for any inputs $x, x' \in \mathbf{R}^{d_0}$ such that $\|x\| \leq 1, \|x'\| \leq 1$, with probability at least $1 - \delta$ we have*

$$\left| \left\langle \frac{\partial f(x,\theta)}{\partial \theta}, \frac{\partial f(x',\theta)}{\partial \theta} \right\rangle - \Theta^{(L)}(x, x') \right| \leq (L + 1)\epsilon \tag{9}$$

Theorem 2. [1, Th 3.2] *Suppose $\sigma(z) = \max(0, z)$, $1/\kappa = poly(1/\epsilon, \log(n/\delta))$ and $d_1 = d_2 = \cdots = d_L = m$ with $m \geq poly(1/\kappa, L, 1/\lambda_0, n, log(1/\delta))$. Then for any $x_{te} \in \mathbf{R}^d$ with $\|x_{te}\| = 1$ with probability at least $1-\delta$ over the random initialization, we have*

$$|f_{nn}(x_{te}) - f_{ntk}(x_{te})| \leq \epsilon \tag{10}$$

These theorems are error analysis for the ReLU activator $\max(0, z)$. As to convergence theorems for other activators, see, e.g., [11, Th 1], [27].

Our definition of the neural network is a composite of linear maps (affine maps without bias terms) and activator functions. Note that when there are bias terms [11], we may set

$$\Sigma^{(0)} = x^T x' + \beta^2 \tag{11}$$

and

$$\Sigma^{(h)}(x, x') = c_\sigma E_{(u,v)\sim N(0,\Lambda^{(h)})}[\sigma(u)\sigma(v)] + \beta^2. \tag{12}$$

Here, β is a hyperparameter.

3 Holonomic activator distribution, HGM, and HIE

Let $\sigma(u)$ be an activator distribution. When it satisfies a linear ordinary differential equation (linear ODE) with polynomial coefficients, it is called *a holonomic activator distribution* or *a holonomic activator function*. Any holonomic activator distribution has finite number of poles as a function on the complex plane, because the pole locus is the zero of the leading coefficient of the ODE. The derivative of a holonomic activator distribution also satisfies a linear ODE with polynomial coefficients.

The ReLU activator distribution $\sigma(u)$ satisfies $(u\partial_u - 1) \bullet \sigma(u) = 0$ ($\sigma(u)$ is annihilated by $u\partial_u - 1$) where $\partial_u = \frac{d}{du}$ and $\bullet$ means the action of a differential operator to a distribution. The derivative $\dot{\sigma}(u)$ is annihilated by $u\partial_u$.

Here is a list of holonomic activator distributions from the list of Wikipedia article of activator functions: binary step, rectified linear unit (ReLU), Gaussian error linear unit (GeLU), exponential linear unit (ELU), scaled exponential linear unit (SELU), Leaky rectified linear unit (Leaky ReLU), parametric rectified linear unit (PReLU), Gaussian. Note that the sigmoid function $\frac{1}{1+e^{-x}}$ is *not* a holonomic activator distribution. Because it has infinitely many poles at $x = \sqrt{-1}(\pi + 2n\pi)$, $n \in \mathbf{Z}$ in the complex plane.

We consider the expectation $E_{(u,v)\sim N(0,\Sigma)}[\sigma(u)\sigma(v)]$ where $N(0, \Sigma)$ is the 2 dimensional normal distribution of the average 0 and the covariance Σ. Put $x = -\frac{1}{2}\Sigma^{-1}$ where x is the matrix $\begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix}$, $x_{12} = x_{21}$. The expectation is written as $g(x)/Z(x)$ where

$$g(x) = \int_{\mathbf{R}^2} \sigma(u)\sigma(v) \exp(x_{11}u^2 + 2x_{12}uv + x_{22}v^2) du dv \tag{13}$$

and

$$Z(x) = \int_{\mathbf{R}^2} \exp(x_{11}u^2 + 2x_{12}uv + x_{22}v^2) du dv = \frac{\pi}{\sqrt{x_{11}x_{22} - x_{12}^2}}. \tag{14}$$

We will call $g(x)$ the unnormalized expectation and we denote the unnormalized expectation by $\hat{E}$ as

$$\hat{E}[\sigma_1(u)\sigma_2(v)] = \int_{\mathbf{R}^2} \sigma_1(u)\sigma_2(v) \exp(x_{11}u^2 + 2x_{12}uv + x_{22}v^2) du dv \tag{15}$$

for random variables $\sigma_1(u)$ and $\sigma_2(v)$. In this paper, we evaluate this $\hat{E}$ as the function of x unlike other literature. The relationship with the expectation value expressed by Σ is

$$E_{(u,v)\sim N(0,\Sigma)}[\sigma_1(u)\sigma_2(v)] = \hat{E}[\sigma_1(u)\sigma_2(v)] \frac{\sqrt{\det(x)}}{\pi}, \quad \Sigma = -\frac{1}{2}x^{-1}. \tag{16}$$

This expectation $E_{(u,v)\sim N(0,\Sigma)}[\sigma_1(u)\sigma_2(v)]$ is often denoted by

$$k_{\sigma_1\sigma_2}(c_1, c_2, r), \quad \Sigma = \begin{pmatrix} c_1^2 & c_1c_2r \\ c_1c_2r & c_2^2 \end{pmatrix}, c_i > 0 \tag{17}$$

to express the dual activation in other literature. When $\sigma_1 = \sigma_2$, $k_{\sigma_1\sigma_2}$ is denoted by k_σ. See, e.g., [8].

Let $D_n = \mathbf{C}\langle x_1, \ldots, x_n, \partial_1, \ldots, \partial_n\rangle$ be the ring of differential operators where $\partial_i = \frac{\partial}{\partial x_i}$. Let $\ell = \sum_{(\alpha,\beta)\in E} c_{\alpha\beta} x^\alpha \partial^\beta$ be an element of D_n where $c_{\alpha\beta} \in \mathbf{C}$, $x^\alpha = \prod_{i=1}^n x_i^{\alpha_i}$, $\partial^\beta = \prod_{i=1}^n \partial_i^{\beta_i}$, and E is a finite subset of $\mathbf{Z}_{\geq 0}^{2n}$. A left ideal I in D_n is called *a holonomic ideal* or *a holonomic system* (of linear PDE's) when the dimension of the zero set of the ideal generated by the principal symbols of I is n. For example, the principal symbol of $x_1\partial_1^2 + 1$ is $x_1\xi_1^2 \in \mathbf{C}[x_1, \xi_1]$ and $\dim V(x_1\xi_1^2) = 1$. Then the left ideal generated by $x_1\partial_1^2 + 1$ in D_1 is a holonomic ideal. See, e.g., [9, 6.4, 6.8] and [22] on the notion of a holonomic ideal. A function (or a distribution) is called a *holonomic function* (or a holonomic distribution) when it is annihilated by a holonomic ideal. The following theorem by I.N.Bernstein [6] is the theoretical foundation of our method.

Theorem 3. *[6], see also, e.g., [9, Th 6.10.8].*
If the left ideal I of D_n is holonomic, then the intersection of the sum of left ideal and right ideal and D_{n-1}

$$(I + \partial_n D_n) \cap D_n \tag{18}$$

is a holonomic ideal in D_{n-1}.

Roughly speaking, the theorem implies that if f is a holonomic function in n variables, then $\int_{\mathbf{R}} f dx_n$ is a holonomic function in $n-1$ variables. An algorithm of construct the *integration ideal* (18) is given by T.Oaku [15] (see also, e.g., [9, Chap 6]).

Let R_n be the rational Weyl algebra (the ring of differential operators with rational function coefficients $\mathbf{C}(x)\langle\partial_1, \ldots, \partial_n\rangle$, $\mathbf{C}(x) = \mathbf{C}(x_1, \ldots, x_n)$). It is known that when I is holonomic, then $r := \dim_{\mathbf{C}(x)} R_n/(R_n I)$ is finite. The dimension r is called the *holonomic rank* of I. The holonomic rank is equal to the dimension of the analytic solutions of I at a generic point. Let $s_1 = 1, s_2, \ldots, s_r$ be a basis of $R_n/(R_n I)$ regarded as a vector space over $\mathbf{C}(x)$. When they are monomials of ∂, they are called *standard monomials*. Then, $\partial_i s_j$ can be expressed as a linear combination of s_k's as $\partial_i s_j = \sum_{k=1}^r p_{jk}^i(x) s_k$ in $R_n/(R_n I)$. The rational functions p_{jk}^i can be obtained by a Gröbner basis computation (see, e.g., [9, 6.1, 6.2]). If a function f is annihilated by the left ideal I, then $F = (f, s_2 \bullet f, \ldots, s_r \bullet f)^T$ satisfies

$$\frac{\partial F}{\partial x_i} = P_i F \tag{19}$$

where P_i is a $r \times r$ matrix $P_i = (p_{jk}^i)$. The equation is called *a Pfaffian system*. It is also expressed as

$$dF = (P_1 dx_1 + \cdots + P_n dx_n) F. \tag{20}$$

It is well-known that an ODE of the rank r and the independent variable z can be translated to a system of first order ODE $\partial_z \bullet F = P(z)F$ where $P(z)$ is $r \times r$ matrix. A Pfaffian system associated to a holonomic system is a generalization of this system. See, e.g., [9, §6.2].

A holonomic gradient method (HGM) utilizing the construction above was introduced in [20] and [21]. It gives an algorithmic method to evaluate normalizing constant and expectations. The HGM is performed by the following 3 steps.

Algorithm 1. HGM ([20], [21], [9, 6.5, 6.11]).

1. *Derive a holonomic ideal and a Pfaffian system satisfied by a definite integral $e(x)$ with parameter x, e.g., $e(x) = \hat{E}[\sigma_1(u)\sigma_2(v)]$.*
2. *Evaluate $e(x)$ and its derivatives at a special point $x = x_0$.*
3. *Solve numerically the Pfaffian system with values obtained in the step 2.*

A difference analogy of the above algorithm is called *difference HGM*, which will be discussed in Section 4.

Our algorithm to evaluate $\hat{E}[\sigma_1(u)\sigma_2(v)]$ follows the general scheme of the HGM, but is more specialized for computing the expectation of holonomic activator distributions. The specialization is based on the following fact by Koyama-Takemura.

Theorem 4. [12, Th 1, 2]
If a tempered distribution $f(t)$ on $\mathbf{R}^d$ is annihilated by $P_1, \ldots, P_s$, then the integral

$$\int_{\mathbf{R}^d} f(t) \exp\left(\sum_{i,j=1}^{d} t_i x_{ij} t_j + \sum_{i=1}^{d} t_i y_i \right) dt_1 \cdots dt_d \tag{21}$$

is annihiated by

$$\varphi(P_k), \qquad 1 \le k \le s, \tag{22}$$

$$\partial_{x_{ij}} - 2\partial_{y_i}\partial_{y_j}, \qquad 1 \le i < j \le d, \tag{23}$$

$$\partial_{x_{ii}} - \partial_{y_i}^2, \qquad 1 \le i \le d. \tag{24}$$

Here, $x_{ij} = x_{ji}$ and $\varphi(t_i) = \partial_{y_i}$ and $\varphi(\partial_{t_i}) = -y_i - 2\sum_{k=1}^{d} x_{ik}\partial_{y_k}$. If the operators $P_1, \ldots, P_s$ generate a holonomic ideal, then (22), (23), (24) generate a holonomic ideal.

Algorithm 2.
Input: Linear ODE ℓ_1 and ℓ_2 annihilating $\sigma_1(u)$ and $\sigma_2(u)$ respectively. A curve on the x space.
Output: Values of $\hat{E}[\sigma_1(u)\sigma_2(v)]$ (15) on a curve.

1. *Apply [12, Th 2] (Theorem 4) to the left ideal generated by ℓ_1 and ℓ_2 in $\mathbf{C}\langle u, v, \partial_u, \partial_v\rangle$ and obtain a holonomic ideal I_1 in $\mathbf{C}\langle x_{11}, x_{12}, x_{22}, y_1, y_2, \partial_{11}, \partial_{12}, \partial_{22}, \partial_1, \partial_y\rangle$.*
2. *Apply a restriction algorithm [15] to find generators of $I_2 := I_1 \cap \mathbf{C}\langle x_{11}, x_{12}, x_{22}, \partial_{11}, \partial_{12}, \partial_{22}\rangle$.*
3. *Translate I_2 to a Pfaffian system.*
4. *Evaluate initial values of F at $x_{11} = -1, x_{12} = 0, x_{22} = -1$ or around this point by the series of Proposition 1.*
5. *Solve the Pfaffian system numerically on a given curve.*

Although the restriction algorithm of the step 2 works for any holonomic input on computer algebra systems in principle, it would be better if the ideal I_2 could be determined by a calculation by hand. In fact, following the steps 1 and 2 of Algorithm 2 by hand, we have the following theorem, which expresses the dual activation in terms of the Gauss hypergeometric function ${}_2F_1(\alpha, \beta, \gamma; z)$.

Theorem 5. *Let m, n are non-negative integers and $Y(u)$ the Heaviside function.*

1. *The integrals $\hat{E}[u^m v^n](x_{11}, x_{12}, x_{22})$ and $\hat{E}[u^m v^n Y(u)Y(v)](x_{11}, x_{12}, x_{22})$ satisfy the GKZ hypergeometric system (see, e.g., [22])*

$$2x_{11}\partial_{11} + x_{12}\partial_{12} + m + 1, \tag{25}$$

$$x_{12}\partial_{12} + 2x_{22}\partial_{22} + n + 1, \tag{26}$$

$$4\partial_{11}\partial_{22} - \partial_{12}^2 \tag{27}$$

2. *Assume $x_{11}, x_{22} < 0$ and $0 \le \frac{x_{12}^2}{x_{11}x_{22}} < 1$. Then, the solution space of the GKZ system above is spanned by*

$$\varphi_1 := (-x_{11})^{-\alpha}(-x_{22})^{-\beta}\, {}_2F_1\left(\alpha, \beta, \frac{1}{2}; z\right) \tag{28}$$

$$\varphi_2 := (-x_{11})^{-\alpha}(-x_{22})^{-\beta}\sqrt{z}\,\mathrm{sign}\,(x_{12})\, {}_2F_1\left(\alpha + \frac{1}{2}, \beta + \frac{1}{2}, \frac{3}{2}; z\right) \tag{29}$$

where

$$\alpha = \frac{1+m}{2}, \beta = \frac{1+n}{2}, z = \frac{x_{12}^2}{x_{11}x_{22}} \tag{30}$$

and $\mathrm{sign}\,(x)$ is the sign of x.

3. *Assume* $x_{11}, x_{22} < 0$ *and* $\frac{x_{12}^2}{x_{11}x_{22}} < 1$. *When* m, n *are even numbers, the integral* $\hat{E}[u^m v^n](x_{11}, x_{12}, x_{22})$ *is equal to* $\Gamma(\alpha)\Gamma(\beta)\varphi_1$. *If both* m, n *are odd numbers, the integral* $\hat{E}[u^m v^n](x_{11}, x_{12}, x_{22})$ *is equal to* $\frac{1}{2}mn\Gamma\left(\alpha - \frac{1}{2}\right)\Gamma\left(\beta - \frac{1}{2}\right)\varphi_2$. *If either* m *or* n *is odd, then the integral is equal to* 0.

4. *Assume* $x_{11}, x_{22} < 0$ *and* $\frac{x_{12}^2}{x_{11}x_{22}} < 1$. *The integral* $\hat{E}[u^m v^n Y(u)Y(v)](x_{11}, x_{12}, x_{22})$ *is equal to*

$$\frac{1}{4}\Gamma(\alpha)\Gamma(\beta)\varphi_1 + \frac{1}{2}\Gamma\left(\alpha + \frac{1}{2}\right)\Gamma\left(\beta + \frac{1}{2}\right)\varphi_2 \tag{31}$$

A proof of this theorem is given in Appendix 8.1. Note that we have

$$ {}_2F_1((1+m)/2, 1/2, 1/2; z) = (1-z)^{-1/2-m/2}, \tag{32}$$

$$ {}_2F_1(1, 1, 1/2; z) = \left(1 + \frac{\sqrt{z}\arcsin(\sqrt{z})}{\sqrt{1-z}}\right)(1-z)^{-1}, \tag{33}$$

$$ {}_2F_1(3/2, 3/2, 3/2; z) = (1-z)^{-3/2} \tag{34}$$

and

$$\begin{aligned} \frac{1}{a}(z\partial_z + a) \bullet {}_2F_1(a, b, 1/2; z) &= {}_2F_1(a+1, b, 1/2; z) &(35)\\ \frac{1}{b}(z\partial_z + b) \bullet {}_2F_1(a, b, 1/2; z) &= {}_2F_1(a, b+1, 1/2; z), &(36)\end{aligned}$$

which are called contiguity relations. These identities give a closed form of (28) when m, n are given. A closed form of the dual activation of a polynomial activator is given by Han et al [8, Theorem 1]. Note that $(\sum_{i=0}^q a_i u^i)(\sum_{j=0}^q a_j v^j) = \sum_{i,j=0}^q a_i a_j u^i v^j$. Then, our theorem gives a different closed form expression of the dual activation for a polynomial activator $\sum_{i=0}^q a_i u^i$. Analogously, our formula gives the dual activation for a rectified polynomial activation $\sigma(u) = \left(\sum_{i=0}^q a_i u^i\right) Y(u)$, because we have

$$\sigma(u)\sigma(v) = \sum_{i,j=0}^{q} a_i a_j u^i v^j Y(u)Y(v). \tag{37}$$

The dual activation for a monomial is given in [8, F.7] by the hypergeometric function ${}_2F_1$. This formula is a special case of our theorem in a different form. A closed from for a rectified monomial is known [7]. Our formula for the rectified polynomial generalizes it and seems to be new as long as we know.

Let us come back to the general algorithm of the HGM. We use the following proposition to perform the step 4.

Proposition 1. *Series expansion of* $\hat{E}[\sigma_1(u)\sigma_2(v)]$ *at* $(x_{11}, x_{12}, x_{22}) = (-1, 0, -1)$ *is* $\sum_{k \in \mathbf{N}_0^3} c_k x^k$, $x^k = x_{11}^{k_{11}} x_{12}^{k_{12}} x_{22}^{k_{22}}$ *where*

$$\begin{aligned} c_k = \frac{2^{k_{12}}}{k_{11}!k_{12}!k_{22}!} &\times \int_{-\infty}^{\infty} u^{2k_{11}+k_{12}} \sigma_1(u)\exp(-u^2)du \\ &\times \int_{-\infty}^{\infty} v^{2k_{22}+k_{12}} \sigma_2(v)\exp(-v^2)dv. \end{aligned} \tag{38}$$

The holonomic system can also be used to obtain an approximate expression of the expectation $\hat{E}[\sigma_1(u)\sigma_2(v)]$ in terms of a set of basis functions by the sparse interpolation and extrapolation method B [23]. We call the following algorithm the *holonomic interpolation/extrapolation* method (HIE).

Algorithm 3.
Input: Linear ODE ℓ_1 and ℓ_2 annihilating $\sigma_1(u)$ and $\sigma_2(u)$ respectively. A set of functions $B = \{e_\beta(x) \,|\, \beta \in \mathcal{B}\}$ on the x space. A numerical integration scheme (t_j, T_j) (evaluation points $\{t_j\}$ and positive weights $\{T_j\}$). γ-th derivative values $\{q_k^{(\gamma)}\}$ of the expectation $\hat{E}[\sigma_1(u)\sigma_2(v)]$ at $\{p_k \,|\, k = 1, \ldots, r\}$ in the x space.
Output: An approximation of $\hat{E}[\sigma_1(u)\sigma_2(v)]$ (15) in terms of the set of functions B.

1. *Apply the same procedure of the first two steps of Algorithm 2.*
2. *Let ℓ_i, $i = 1, \ldots, s$ be generators of the left ideal I_2.*
3. *Put $f(x) = \sum_{\beta \in \mathcal{B}} f_\beta e_\beta(x)$ where f_β are unknown coefficients.*
4. *Minimize*
$$\ell(\{f_\beta\}) := \sum_{i=1}^{s} \sum_j T_j \left| \sum_{\beta \in \mathcal{B}} f_\beta (\ell_i e_\beta)(t_j) \right|^2 \tag{39}$$
under the constraints at data points
$$\sum_\beta f_\beta e_\beta^{(\gamma)}(p_k) = q_k^{(\gamma)}, \quad k = 1, \ldots, r, \alpha \in \Gamma \tag{40}$$
where $e_\beta^{(\alpha)}$ is $\partial_\alpha \bullet e_\beta$ (γ-th derivative of e_β).
5. *Return $\sum_\beta f_\beta e_\beta(x)$.*

We give remarks about this algorithm.

Since the constraints are linear, we can parametrize the space of f_β's by an affine map and reduce the problem to a least square problem with no constraint. Or, the loss function for the minimization may be set as
$$\ell(\{f_\beta\}) := \sum_{i=1}^{s} \sum_j T_j \left| \sum_{\beta \in \mathcal{B}} f_\beta (\ell_i e_\beta)(t_j) \right|^2 + \mu \sum_{k=1}^{r} \left| \sum_\beta f_\beta e_\beta(p_k) - q_k \right|^2 \tag{41}$$
to transform the minimization problem to that with no constraint. It is also a least square problem. Here, μ is a paramter. The larger the paramter μ, the closer the solution is to the given values of f at $\{p_k\}$.

There are various ways to select a set of basis functions. For example, if we can find a set of fundamental solutions of $I_2 \bullet e_\beta = 0$, the problem is reduced to find a best set of coefficients f_β satisfying the constraints (40) approximately. In particular, if the basis functions are a basis of series solutions at a point p_k and given values are derivatives standing for standard monomials at $x = p_k$, then our algorithm constructs the series expansion of the expectation at $x = p_k$.

Finally, we briefly note some other applications of our holonomic approach. Holonomic systems for $\hat{E}$ can be utilized to derive several formulas of the function $\hat{E}$ other than obtaining series expressions. For example, it can be used in the following ways; finding a higher order ODE for one direction (e.g., [9, Th 6.1.11]), estimating an asymptoric expansion of $\hat{E}$ at a singular point (e.g., [5]), finding a rational solution (e.g.,[3]).

4 Algorithms to derive an Hermite expansion for a holonomic activation function

Han et al [8, Th 2] gave a method to evaluate the expectation $\hat{E}[\sigma(u)\sigma(v)]$ by utilizing the Hermite expansion of $\sigma(u)$. Let $He_n(t)$ be probabilist's n-th Hermite polynomial e.g., $He_0(t) = 1$, $He_1(t) = t$, $He_2(t) = t^2 - 1$, $He_3(t) = t^3 - 3t$, ... The n-th coefficient of the Hermite expansion of $\sigma(u)$ is $\frac{c_n}{\sqrt{2\pi n!}}$ where
$$c_n = \int_{-\infty}^{\infty} \sigma(u) He_n(u) \exp(-u^2/2) du. \tag{42}$$

The function $\sigma(u)$ is expressed as $\sum_{n=0}^{\infty} \frac{c_n}{\sqrt{\pi} n!} He_n(x)$. If $\sigma(u)$ is a holonomic function, then c_n satisfies a linear difference equation. Creative telescoping algorithm (see, e.g., [18], [13]) or the integration algorithm of D-modules with the Mellin transformation (see, e.g., [16]) can be used to derive it. By *difference HGM*, we mean that obtaining c_n by the difference equation of rank r (recurrence relation) and initial values $c_0, \ldots, c_{r-1}$.

Examples of deriving Hermite expansions by these algorithms are given in Section 6.1.3 and Appendix 8.5.

The Hermite expansion expresses the dual activation as follows. These results are by [4] and [8].

Theorem 6. [4], [8]

1. [4] *If σ is absolutely continuous and satisfies a homogenity $\sigma(at) = |a|^q \sigma(t)$ for all $a, t \in \mathbf{R}$, the the dual activation is*
$$k_\sigma(c_1, c_2, r) = c_1^q c_2^q \sum_{j=0}^{\infty} \left(\frac{c_j}{\sqrt{\pi} j!} \right)^2 r^j \tag{43}$$

2. [8] *Let $\sigma(t)$ be a polynomial $\sum_{j=0}^{q} a_j t^j$. The dual activation is*
$$k_\sigma(c_1, c_2, r) = \sum_{\ell=0}^{q} r_\ell(c_1) r_\ell(c_2) r^\ell \tag{44}$$
where
$$r_\ell(t) = \sum_{i=0}^{\lfloor (q-\ell)/2 \rfloor} \frac{a_{\ell+2i} (\ell+2i)!}{2^i i! \sqrt{\ell!}} t^{2i+\ell}. \tag{45}$$

As we will see later, the difference HGM is more efficient than evaluating the integral (42) indivisually. Thus, the difference HGM strengthens the Hermite expansion method above.

5 Faster Evaluation by the HGM

"HGM all at once method". When a definite integral with parameters satisfies a holonomic system, it is possible to find integral values at many parameter points by a single run of the Runge-Kutta method. It is an advantage of utilizing the HGM.

Let us explain what it means by an example. We use a Pfaffian system given in [9, §6.2] for our example. Put $P_1 = \begin{pmatrix} 0 & z_2/z_1 \\ -z_1 z_2 & 1/z_1 \end{pmatrix}$ and $P_2 = \begin{pmatrix} 0 & 1 \\ -z_1^2 & 0 \end{pmatrix}$. Consider the Pfaffian system

$$\partial_{z_1} \bullet F = P_1 F, \ \partial_{z_2} \bullet F = P_2 F$$

where $F = (1, \partial_{z_2})^T \bullet f$. Note that $f = -\int_{\pi/2}^{z_1 z_2} \sin(t) dt = \cos(z_1 z_2)$ is a solution. Suppose that we want to evalute f at $p_1 = (\pi/2, 1)$, $p_2 = (\pi/2, 2)$ and $p_3 = (\pi, 3)$. Let $p_0 = (z_1, z_2) = (\pi/2, 0)$ be the starting point of the Runge-Kutta method. The value of F at the point is $(1, 0)$. We apply the Runge-Kutta method along the piecewise linear path connecting p_0, p_1, p_2, p_3. In other words, we solve the ODE

$$\frac{dF}{dt} = P_2(\pi/2, t) F \quad \text{for } t \in [0, 2]$$

to find values of F at p_1 and p_2, and solve the ODE

$$\frac{dF}{dt} = \left(P_1(z_1(t), z_2(t)) \frac{dz_1(t)}{dt} + P_2(z_1(t), z_2(t)) \frac{dz_2(t)}{dt} \right) F$$

along the path $z_1(t) = \pi/2 + (\pi - \pi/2)(t-2)$, $z_2(t) = 2 + (3-2)(t-2)$, $t \in [2,3]$ with the initial condition $F(p_2)$ to obtain the value F at p_3. It is faster than applying the Runge-Kutta method 3 times independently from p_0 to p_i, $i = 1,2,3$. See [17] as to a sample code.

"Taylor expansion with HGM" method. Other approach to accelerate the HGM is to solve an ODE on a curve or a line and determine the value of f near a point of the curve or the line by a Taylor expansion. We explain this method by the example above. We denote by f_{ij} the derivative $\frac{\partial^{i+j} f}{\partial z_1^i \partial z_2^j}$. Since $\partial_{z_1} \bullet F = (f_{10}, f_{11})$, we can obtain the value of (f_{10}, f_{11}) by evaluating $P_1 F$. Note that $F = (f, f_{01})$. Let a be a point on the curve or the line. Since the first order Taylor expansion at $z = a$ is $f(a+h) = f(a) + f_{10}(a)h_1 + f_{01}(a)h_2$, we can express $f(a+h)$ in terms of the value of F at $z = a$. Values of higher order derivatives can also be expressed by F by differentiating the Pfaffian system. For example, we have $\partial_{z_1}^2 \bullet F = \frac{\partial P_1}{\partial z_1} F + P_1 \frac{\partial F}{\partial z_1} = \frac{\partial P_1}{\partial z_1} F + P_1^2 F$ and $\partial_{z_1}^2 \bullet F = (f_{20}, f_{21})$. Then, $f_{20}(a)$ and $f_{21}(a)$ can be expressed in terms of $F(a)$.

6 HGM and HIE for ReLU and some other activator functions

6.1 ReLU

Let $\sigma(u)$ be ReLU (rectified linear unit) function; $\sigma(u) = \max(u, 0) = uY(u)$ where $Y(u)$ is the Heaviside function. Closed forms of the expectations (1) and (2) in terms of arccos and $\sqrt{\ }$ are known for the activator function ReLU. Let Λ be $\begin{pmatrix} c_1^2 & c_1 c_2 r \\ c_1 c_2 r & c_2^2 \end{pmatrix}, c_1, c_2 \geq 0, |r| \leq 1$ Then,

$$E_{(u,v)\sim N(0,\Lambda)}[\sigma(u)\sigma(v)] = \frac{r(\pi - \arccos(r)) + \sqrt{1-r^2}}{2\pi} \cdot c_1 c_2 \tag{46}$$

$$E_{(u,v)\sim N(0,\Lambda(h))}[\dot{\sigma}(u)\dot{\sigma}(v)] = \frac{\pi - \arccos(r)}{2\pi} \tag{47}$$

See, e.g., [7], [1, Appendix I] as to details. By specializing our Theorem 5 to the case $m = n = 1$, we also obtain a closed form in a different form. Although closed forms are already known, we explain the algorithmic procedure of the HGM and HIE by using the example of the ReLU, because a small example is easier to understand the concept of our method.

6.1.1 HGM for ReLU

Since ReLU $\sigma(u)$ satisfies the linear differential equation $(u\partial_u - 1) \bullet \sigma(u) = uY'(u) + uY(u) - \sigma(u) = 0$ $(uY'(u) = u\delta(u) = 0)$ as the distribution, we can apply [12, Th 2] (Theorem 4) to obtain *a holonomic system* satisfied by $g(x)$. See Appendix 8.1 as to details. The steps 1 and 2 of Algorithm 2 are performed by hand there. See Appendix 8.2 to perform the steps 1 and 2 by a computer algebra system.

Applying step 3 of Algorithm 2, we obtain the following theorem by Gröbner basis computations.

Theorem 7. *1. The holonomic system (66) is of rank 2. Put*

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

Then we have the Pfaffian system $\partial_{x_{11}} \bullet F - P_{11} F = 0, \partial_{x_{12}} \bullet F - P_{12} F = 0, \partial_{x_{22}} \bullet F - P_{22} F = 0$. Explicit form of the 2×2 matrix P_{ij} is given in Appendix 8.3.

2. The singular locus of the Pfaffian system (the denominator of the matrices P_{ij}) is

$$x_{11} x_{22} (x_{12}^2 - x_{22} x_{11}). \tag{48}$$

Note that the condition that $-X$ is positive definite is

$$x_{11} < 0, \quad x_{11} x_{22} - x_{12}^2 > 0. \tag{49}$$

Let us proceed on the step 4 of Algorithm 2. We apply Proposition 1. We take the special point $x = (x_{11}, x_{12}, x_{22}) = (-1, 0, -1) =: x_0$ where the integral splits to a product of single integrals;

$$\begin{aligned} & \partial_{x_{11}}^{d_{11}} \partial_{x_{12}}^{d_{12}} \partial_{x_{22}}^{d_{22}} \bullet g(x)|_{x=x_0} \\ = \; & 2^{d_{12}} \int_0^\infty u^{1+2d_{11}+d_{12}} \exp(-u^2) du \int_0^\infty v^{1+d_{12}+2d_{22}} \exp(-v^2) dv. \end{aligned} \tag{50}$$

Since

$$\int_0^\infty u^m \exp(-u^2) du = \frac{\Gamma\left(\frac{1+m}{2}\right)}{2}, \tag{51}$$

$\Gamma(n+1) = n!$, and $\Gamma\left(\frac{1}{2} + n\right) = \frac{(2n-1)!!}{2^n}\sqrt{\pi}$, we have the series expansion of $g(x)$ at $x = x_0$ as

$$\begin{aligned} g(x) &= \sum_{d_{12}:\text{even}} \frac{2^{d_{12}}}{4d!} (d_{11} + d_{12}/2)!(d_{22} + d_{12}/2)! {x'_{11}}^{d_{11}} x_{12}^{d_{12}} {x'_{22}}^{d_{22}} \\ &+ \pi \sum_{d_{12}:\text{odd}} \frac{2^{d_{12}}(2d_{11} + d_{12})!!(2d_{22} + d_{12})!!}{2^{3+d_{11}+d_{22}+d_{12}} d!} {x'_{11}}^{d_{11}} x_{12}^{d_{12}} {x'_{22}}^{d_{22}} \end{aligned} \tag{52}$$

where $d! = d_{11}!d_{12}!d_{22}!$, $x'_{ij} = x_{ij} + 1$, and $\sum_{d_{12}:\text{even}}$ means $\sum_{\{d \in \mathbf{N}_0^3 \,|\, d_{12}:\text{even}\}}$. In particular, we have

$$(1, \partial_{x_{12}}) \bullet g(x)|_{x=x_0} = \left(\frac{1}{4}, \frac{\pi}{8}\right). \tag{53}$$

Let us perform the step 5 of Algorithm 2. The domain $x_{11} < 0, x_{11}x_{22} - x_{12}^2 > 0$ is convex. Choose a point x_1 in the domain. Since the domain is convex, we can restrict the Pfaffian system on the line $rt + x_0$, $r = x_1 - x_0$, $0 \le t \le 1$ and obtain the ODE

$$\frac{dF}{dt} = \left(P_{11}(rt + x_0)r_{11} + P_{12}(rt + x_0)r_{12} + P_{22}(rt + x_0)r_{22}\right) F, \tag{54}$$

$$F(0) = \left(\frac{1}{4}, \frac{\pi}{8}\right)^T. \tag{55}$$

The fist element of $F(1)$ gives the value of $g(x)$ at $x = x_1$.

An evaluation for $\dot{\sigma}$ by the HGM is explained in Appendix 8.4.

6.1.2 HIE for ReLU

We restrict the system of differential equation to $x_{11} = x_{22} = -1$ and apply the HIE. The ODE on $x_{11} = x_{22} = -1$ is

$$L \bullet f = 0, \quad L = (1 - x_{12}^2)\partial_{12}^2 - 5x_{12}\partial_{12} - 4, \tag{56}$$

which can be obtained by the restriction algorithm. We want to solve it on $x_{12} \in [-1, 1]$. Let $e_0, e_1, \ldots, e_m$ be a basis and suppose that $f = \sum_{\beta=0}^m f_\beta e_\beta$ where f_β's are unknown coefficients. We suppose that the value of f is given at p_1 and p_2 and the correponding values are q_1 and q_2. Then, the loss function is

$$\sum_{j=0}^n T_j \left(f_0(L \bullet e_0)(t_j) + f_1(L \bullet e_1)(t_j) + \cdots f_m(L \bullet e_m)(t_j)\right)^2 + \mu \sum_{k=1}^2 \left(\sum_{\beta=0}^m f_\beta e_\beta(p_k) - q_k\right)^2 \tag{57}$$

Since $(L \bullet e_\beta)(t_j)$'s and $e_\beta(p_k)$'s are numbers, it is a least square problem for an unknown vector $(f_0, f_1, \ldots, f_m)$.

6.1.3 Hermite expansion of ReLU

Let $\sigma(u) = Y(u)u$ be the ReLU function where $Y(u)$ is the Heaviside function. It satisfies $(u\partial_u - 1)\bullet\sigma(u) = 0$ as a distribution. Apply an algorithm to find annihilating ideal for a product of distributions (see, e.g., [16]), we find an annihilating operator for $Y(u)u \cdot He_n(u)\exp(-u^2/2)$. It also annihilated by the difference operator $S_n^2 - uS_n + (n+1)$ where S_n is the shift operator for the variable n. For example, we have $S_n \bullet He_n = He_{n+1}$. Applying an algorithm to find linear difference equation for c_n we obtain a difference operator annihilating c_n as

$$s_n^2 + (n-1)$$

Initial values are $c_0 = 1$, $c_1 = \sqrt{\frac{\pi}{2}}$.

6.2 Other activator distributions

The HGM can be applied for any holonomic activator distributions. To demonstrate this fact, we discuss on the HGM for an activator function

$$Y(u)\sin u, \tag{58}$$

which we will call ReSin, in Appendix 8.5. Note that ReSin is not a smooth function. We also discuss on GeLU in Appendix 8.6, which is a smooth function. Note that the Hermite expansion by [8] provides very low approximate errors for smooth activator functions like GeLU[1], but not for non-smooth distributions like ReLU and ReSin $Y(u)\sin u$.

7 Experiments

In this section, we perform experiments on our algorithms. Before showing data of experiments, we explain what we mean by learning and inference. Let $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ be training data. Here x_i is an input and y_i is an output. Let $K(x, x')$ be the kernel function. The learning in the kernel method means that obtaining a matrix $H^* \in \mathbf{R}^{N\times N}$ whose i, j element is $K(x_i, x_j)$ from the training data. We call H^* the *kernel matrix*. We mean by inference obtaining an output from any input x by the map

$$f(x) = (K(x, x_1), K(x, x_2), \ldots, K(x, x_N))(H^* + \lambda E)^{-1}(y_1, y_2, \ldots, y_N)^T. \tag{59}$$

Here E is the identity matrix and the term λE is added to H^* because there are cases where H^* does not have the inverse. We put $\lambda = 0.01$ in our experiments. The kernel function $K(x, x')$ is approximated by the function Θ.

Example 1. Learning $\sin \pi x$ by a 2 layer neural network with bias terms. Input and output are 1 dimensional. Training data (x_i, y_i) are equally spaced 15 points x_i's in $[-1, 1]$ and their $y_i = \sin(\pi x_i)$'s values. Inference uses equally spaced 20 points in $[-1, 1]$.

This small problem is our running example to make experiments by our methods for some activators.

7.1 Comparison of our methods for some activators

We compare the following methods for our running example 1.

1. Closed forms of dual activation. It is referred as "closed".
2. Gauss-Hermite quadrature of [8, 3.3]. It is referred as "GaussHerm" or "gh". We use an adaptive meshsize control with the relative error tolerance `1e-10`.
3. HGM of Algorithm 2. It is referred as "hgm". Our implementation uses `scipy.solve_ivp` function with `rtol=1e-10`.

[1]Note that closed form of the expectations for GeLU is given in [25] and [8].

4. HGM all at once given in Section 5. It is referred as "all-at-once" or "aao".

Note that these methods except closed forms work for any holonomic activator functions. Monte-Carlo methods also work for any holonomic activator functions, but Monte-Carlo methods are relatively slow and inaccurate, and then we do not make a comparison.

Timing data are taken on a machine with Intel Core i5-12400 CPU (4.4 GHz). We use numpy 1.24.4 and scipy 1.10.1 on wsl[2].

7.1.1 ReLU

Method	Training time (s)	Inference time (s)
closed	0.007292	
GaussHerm	1.500	1.442
hgm	1.316	1.953
all-at-once	4.352	5.143

	Kernel error
gh − hgm	0.0010347
gh − aao	0.0010348
hgm − aao	2.7771×10^{-8}

	Inference error
GaussHerm	0.97729
hgm	0.96766
all-at-once	0.97460

Here, the kernel error is the Frobenius norm divided by the number of elements of the difference of two kernel matrices. The inference error is the mean square error between the inference values and $\sin \pi x$ values at the 20 points. Note that we use the Ridge regression with $\lambda = 0.01$ and then the graph is a little different with that of $y = \sin \pi x$. As is remarked in [8, Th 2, 3.3], the Gauss-Hermite quadrature or Hermite expansion provide much lower approximation errors than non-smooth ones. Since ReLU is not smooth, the inference by GaussHerm has a little larger inference error. See Figure 1.

7.1.2 GeLU

Method	Training time (s)	Inference time (s)
closed	Not yet in our code	Not yet in our code
GaussHerm	41.25	62.56
hgm	86.21	120.2
all-at-once	32.89	381.1

	Kernel error
gh − hgm	1.1718×10^{-8}
gh − aao	1.4983×10^{-8}
hgm − aao	1.5138×10^{-8}

	Inference error
GaussHerm	0.97163
hgm	0.97166
all-at-once	0.97163

The Gauss-Hermite quadrature provides low approximation errors and the inference error of it is also small, because GeLU is a smooth activator. The HGM and HGM all-at-once provide also low approximation errors as good as GaussHerm. See also Figure 2. Note that our implementation of HGM has not yet included a code to evaluate solutions near the singularity of an ODE, and then we use GaussHerm when det(covariance matrix) $\leq 1 \times 10^{-3}$. Note also that the closed form for GeLU is given in [8], but the case of $\dot{\sigma}$ has not been implemented in our code. It is the reason of "not yet in our code" in the table.

[2]We call this machine "sw".

7.1.3 ReSin

Method	Training time (s)	Inference time (s)
closed	NA	NA
GaussHerm	3.916	4.949
hgm	289.5	1005
all-at-once	21.07	23.39

	Kernel error
gh − hgm	0.0019103
gh − aao	0.0016427
hgm − aao	0.00062839

	Inference error
GaussHerm	0.97328
hgm	0.96874
all-at-once	0.95745

Since ReSin is not smooth, the Gauss-Hermite quadrature does not give a low approximation error. On the other hand, the HGM and HGM all-at-once give a good approximation and HGM all-at-once works in a reasonable time. Starting point of the HGM is $(-1, 1/100, -1)$. Note that the HGM is used only when $1 \times 10^{-3} \leq \det(\text{covariance matrix}) \leq 1$ and the Gauss-Hermite quadrature is used in other intervals because of the same reason of the case of GeLU. A closed form for ReSin is not known except an infinite series expression in terms of contiguous family of Gauss hypergeometric functions (Theorem 5) as long as we know, then the entry of closed form is "NA" (not available).

7.2 Improvements of numerical solvers

We evaluate the unnormalized expecation $\hat{E}$ (15) for the ReSin activator for 91 points (x, x'), which were used when calculating NTK for Example 1. We use the "HGM all at once" method in Section 5 for 91 points. Although there are more points $15^2 - 15 = 210$ to be evaluated, we remove points near the singular locus of the ODE and groups clusters of nearby points into one point to obtain the 91 points. Precisely, we take points satisfying $0.01 \leq \det(\text{covariance matrix}) \leq 1$. Two points whose distance is less than 1×10^{-5} are represented by one point.

We will see that the closer the path of integration of an ODE solver is to a straight line, the faster the HGM will be. Points are extracted for each "step" from among the sorted 91 points. We devide equally segments between these points and create new 91 points. When "step" is 1, the set of new 91 points is nothing but the original one. When "step" increases, the set of points has a distribution closer to a straight line. See Figure 4. The execution time of evaluating $\hat{E}$ at 91 points becomes faster when "step" increases.

step	1	2	5	10	15	20
time (s)	2.065	1.820	1.550	1.115	0.771	0.692

See Figure 5. Note that evaluations at nearby points of a cluster can be done by the Taylor expansion with HGM method in Section 5 from the representative point of the cluster.

The timing data above is taken on AMD EPYC 7552 48-Core Processor of 1.5GHz with 1T bytes memory without GPU [3]. We use the ODE solver `gsl_odeiv` rkf45 with `rtol=1e-10` of the GNU scientific library 2.7.1 written in the language C on the Debian GNU linux 12.2. The program is compiled by gcc version is 12.2.0 with the option -O3. Note that the execution time of the C code is about 1.9 times faster than a python code using the scipy `solve_ivp` with `rtol=1e-10` for the activator ReSin. More precisely, the python code took 3.9166s with scipy version 1.10.1 and the C code took 2.065s.

Acknowledgments
The second author is supported in part by the JST CREST Grant Number JP19209317 and by JSPS KAKENHI Grant Number JP21K03270.

[3] We refer this machine as "machine o3n".

8 Appendix

8.1 Proof of Theorem 5

The function $t_1^m t_2^n$ is annihilated by $t_1\partial_{t_1} - m$, and $t_2\partial_{t_2} - n$. The distribution $t_1^m t_2^n Y(t_1)Y(t_2)$ is also annihilated by these operators becuase $t_1\partial_{t_1} \bullet Y(t_1) = t_1\delta(t_1) = 0$. Applying [12, Th 1, 2], we have the following annihilating operators for the integral $\hat{E}[u^m v^n]$:

$$\partial_1(-y_1 - 2(x_{11}\partial_1 + x_{12}\partial_2)) - m, \tag{60}$$
$$\partial_2(-y_2 - 2(x_{12}\partial_1 + x_{22}\partial_2)) - n, \tag{61}$$
$$\partial_{12} - 2\partial_1\partial_2, \tag{62}$$
$$\partial_{11} - \partial_1^2, \tag{63}$$
$$\partial_{22} - \partial_2^2, \tag{64}$$

where $\partial_{ij} = \partial/\partial x_{ij}$, $\partial_i = \partial/\partial y_i$. Let I_1 be the left ideal generated by the operators above. We want to find elements of $I_2 = (I_1 + y_1 D + y_2 D) \cap \mathbf{C}\langle x_{11}, x_{12}, x_{22}, \partial_{11}, \partial_{12}, \partial_{22}\rangle$ where $D = \mathbf{C}\langle y_1, y_2, x_{11}, x_{12}, x_{22}, \partial_1, \partial_2, \partial_{11}, \partial_{12}, \partial_{22}\rangle$. Expanding (60), we have

$$\begin{aligned} & -y_1\partial_1 - 1 - 2(x_{11}\partial_1^2 + x_{12}\partial_1\partial_2) - m \\ \to\ & -y_1\partial_1 - 2(x_{11}\partial_{11} + (1/2)x_{12}\partial_{12}) - m - 1, \quad \text{by (62) and (63).} \end{aligned}$$

Thus, $-2x_{11}\partial_{11} - x_{12}\partial_{12} - m - 1$ is an element of I_2. Analogusly, we can see that $-x_{12}\partial_{12} - 2x_{22}\partial_{22} - n - 1$ is an element of I_2 from (61). Finally, we have

$$\begin{aligned} & (\partial_{12} - 2\partial_1\partial_2)^2 \\ = \ & 4\partial_1^2\partial_2^2 - 4\partial_1\partial_2\partial_{12} + \partial_{12}^2 \\ \to \ & 4\partial_{11}\partial_{22} - 2\partial_{12}^2 + \partial_{12}^2 \quad \text{by (62)} \sim \text{(64)}, \\ = \ & 4\partial_{11}\partial_{22} - \partial_{12}^2 \end{aligned}$$

We have proved 1. Note: Changing variables $x_{11} \to 2x_{11}$ and $x_{22} \to 2x_{22}$, we obtain the GKZ hypergeometric system of a standard form for the matrix $A = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix}$. We can also obtain the GKZ system for the unnormalized expectation $\hat{E}$ by a theory of integral representations of the GKZ system [14].

Once the system of equations is expressed as a GKZ system, we can apply a general procedure to obtain a series solution, see, e.g., [22]. A solution is written as

$$x_{11}^{\rho_{11}} x_{12}^{\rho_{12}} x_{13}^{\rho_{22}} f(z) \tag{65}$$

where $z = \frac{x_{12}^2}{x_{11}x_{22}}$, $\rho_{11} = -(m+1)/2$, $\rho_{12} = 0$, $\rho_{22} = -(n+1)/2$, and $f(z)$ is a solution of the Gauss hypergeometric differential equation

$$[\theta_z(\theta_z + 1/2 - 1) - z(\theta_z + (m+1)/2)(\theta_z + (n+1)/2)] \bullet f = 0, \quad \theta_z = z\partial_z.$$

It has two independent solutions ${}_2F_1(\alpha, \beta, 1/2; z)$ and $z^{1/2}{}_2F_1(\alpha+1/2, \beta+1/2, 3/2; z)$. Thus, we have proved the statement 2.

Note that the integral $\hat{E}[u^m v^n]$ and $\hat{E}[u^m v^n Y(u)Y(v)]$ are holomorphic at $x_{11} = x_{22} = -1, x_{12} = 0$. Restricting $x_{11} = x_{22} = -1$, we have a series expansion of $c_1\varphi_1 + c_2\varphi_2 = c_1 + c_2 x_{12} + O(x_{12}^2)$. Note that we

have

$$\begin{aligned}
\hat{E}[u^m v^n](-1,0,-1) &= \int_{-\infty}^{\infty} u^m \exp(-u^2)dv \int_{-\infty}^{\infty} v^n \exp(-v^2)dv \\
&= \frac{(1+(-1)^m)(1+(-1)^n)}{4}\Gamma(\alpha)\Gamma(\beta), \\
\hat{E}[u^m v^n Y(u)Y(v)](-1,0,-1) &= \int_{0}^{\infty} u^m \exp(-u^2)dv \int_{0}^{\infty} v^n \exp(-v^2)dv \\
&= \frac{1}{4}\Gamma(\alpha)\Gamma(\beta)
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial \hat{E}[u^m v^n]}{\partial x_{12}}(-1,0,-1) &= 2\int_{-\infty}^{\infty} u^{m+1} \exp(-u^2)dv \int_{-\infty}^{\infty} v^{n+1} \exp(-v^2)dv \\
&= 2\frac{(1+(-1)^{m+1})(1+(-1)^{n+1})}{4} mn\Gamma(\alpha-1/2)\Gamma(\beta-1/2), \\
\frac{\partial \hat{E}[u^m v^n Y(u)Y(v)]}{\partial x_{12}}(-1,0,-1) &= 2\int_{0}^{\infty} u^{m+1} \exp(-u^2)dv \int_{0}^{\infty} v^{n+1} \exp(-v^2)dv \\
&= \frac{1}{2}\Gamma(\alpha+1/2)\Gamma(\beta+1/2).
\end{aligned}$$

The constants c_1, c_2 are determined by these values and we obtain the statements 3 and 4.

8.2 Deriving a holonomic system by computer algebra

We apply the restriction algorithm (see, e.g., [9, §6.10] or [15]) and its implementation on Risa/Asir [19] to the left ideal I generated by

$$\begin{aligned}
&\partial_{y_1}(-y_1 - 2x_{11}\partial_{y_1} - 2x_{12}\partial_{y_2}) - 1, \\
&\partial_{y_2}(-y_2 - 2x_{12}\partial_{y_1} - 2x_{22}\partial_{y_2}) - 1, \\
&\partial_{12} - 2\partial_{y_1}\partial_{y_2}, \\
&\partial_{11} - \partial_{y_1}^2,\ \partial_{22} - \partial_{y_2}^2
\end{aligned}$$

in the ring of differential operators $D = \mathbf{Q}\langle x_{11}, x_{12}, x_{22}, y_1, y_2, \partial_{11}, \partial_{12}, \partial_{22}, \partial_{y_1}, \partial_{y_2}\rangle$. The following ideal called the restriction ideal of I to $y_1 = y_2 = 0$:

$$I' := (I + y_1 D + y_2 D) \cap \mathbf{Q}\langle x_{11}, x_{12}, x_{22}, \partial_{11}, \partial_{12}, \partial_{22}\rangle \tag{66}$$

where $\partial_{ij} = \partial_{x_{ij}}$ These constructions are based on Gröbner bases computation in the Weyl algebra. Here is a Risa/Asir code to obtain I'.

```
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]);
```

8.3 Proof of Theorem 7

We translate I' to a Pfaffian system by a Gröbner basis computation in the ring of differential operators with rational function coefficients (the rational Weyl algebra). See, e.g., [9, §6.2] on the translation. Here is a Risa/Asir code to translate I' to a Pfaffian system.

```
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);
```

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

8.4 HGM for the Derivative of ReLU (Heaviside Function)

Let $\sigma(u)$ be ReLU (rectified linear unit) function; $\sigma(u) = \max(u, 0)$. The derivative of $\sigma(u)$ is the Heaviside function $Y(u)$. Put

$$g(x) = \int_{\mathbf{R}^2} Y(u)Y(v)\exp(x_{11}u^2 + 2x_{12}uv + x_{22}v^2)dudv.$$

The expectation for the Heaviside function $g(x)/Z(x)$ is called *the orthant probability*. Koyama and Takemura [12] gave a method to evaluate it in general dimensions by the HGM. In particular, they show that the 2-dimensional orthant probability satisfies a Pfaffian system of rank 4. Since the average of the normal distribution we consider is 0, the orthant probability satisfies a simpler Pfaffian system than the system given by them. It follows from [12, Th 1, Th 2] that $g(x)$ is annihilated by the left ideal

$$I' = I \cap \mathbf{R}\langle x_{11}, x_{12}, x_{22}, \partial_{11}, \partial_{12}, \partial_{22}\rangle \tag{67}$$

where I is generated by

$$
\begin{aligned}
&\partial_{y_1}(-y_1 - 2x_{11}\partial_{y_1} - 2x_{12}\partial_{y_2}), \\
&\partial_{y_2}(-y_2 - 2x_{12}\partial_{y_1} - 2x_{22}\partial_{y_2}), \\
&\partial_{12} - 2\partial_{y_1}\partial_{y_2}, \\
&\partial_{11} - \partial_{y_1}^2,\ \partial_{22} - \partial_{y_2}^2.
\end{aligned}
$$

Theorem 8. *1. The holonomic system (67) is of rank* 2.

2. Put

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

Then we have the Pfaffian system $\partial_{x_{11}} \bullet F - P_{11}F = 0, \partial_{x_{12}} \bullet F - P_{12}F = 0, \partial_{x_{22}} \bullet F - P_{22}F = 0$ *where*

$$\begin{aligned} d_1 &= -x_{12}^2 + x_{22}x_{11}, \\ P_{11} &= \begin{pmatrix} \frac{-1/2}{x_{11}} & \frac{-1/2x_{12}}{x_{11}} \\ \frac{-1/2x_{12}}{d_1 x_{11}} & \frac{-1/2x_{12}^2 - x_{22}x_{11}}{d_1 x_{11}} \end{pmatrix}, \\ P_{12} &= \begin{pmatrix} 0 & 1 \\ \frac{1}{d_1} & \frac{3x_{12}}{d_1} \end{pmatrix}, \\ P_{22} &= \begin{pmatrix} \frac{-1/2}{x_{22}} & \frac{-1/2x_{12}}{x_{22}} \\ \frac{-1/2x_{12}}{d_1 x_{22}} & \frac{-1/2x_{12}^2 - x_{22}x_{11}}{d_1 x_{22}} \end{pmatrix}. \end{aligned}$$

By (51) and an analogous discussion with ReLU case, We have $(1, \partial_{x_{12}} \bullet g(x)|_{x=x_0} = \left(\frac{\pi}{4}, \frac{1}{2}\right)$, which gives an accurate initial value to solve ODE's for the HGM.

8.5 HGM and difference HGM for ReSin

We call the function $\sigma(y) = Y(u)\sin(u)$ the ReSin (*rectified sine*) function where $Y(x)$ is the Heaviside function. Note that $\sigma(u)$ is not a differentiable function and is a tempered distribution.

8.5.1 HGM for ReSin

Since ReSin $\sigma(u)$ satisfies the linear differential equation $u^2(\partial_u^2 + 1) \bullet \sigma(u) = 0$ as the distribution, we can apply [12, Th 2] (Theorem 4) to obtain *a holonomic system* satisfied by $g(x)$.

Applying steps 2 and 3 of Algorithm 2, we obtain the following theorem by Gröbner basis computations. It took 13.114s on the machine o3n.

Theorem 9. *1. The holonomic system for ReSin of the form (66) is of rank* 8 *and standard monomials can be taken as*

$$(1, \partial_{11}, \partial_{12}, \partial_{22}, \partial_{11}^2, \partial_{12}^2, \partial_{22}^2, \partial_{11}\partial_{12}\partial_{22}). \tag{68}$$

Put

$$F = (1, \partial_{11}, \partial_{12}, \partial_{22}, \partial_{11}^2, \partial_{12}^2, \partial_{22}^2, \partial_{11}\partial_{12}\partial_{22})^T \bullet g.$$

Then we have the Pfaffian system $\partial_{x_{11}} \bullet F - P_{11}F = 0, \partial_{x_{12}} \bullet F - P_{12}F = 0, \partial_{x_{22}} \bullet F - P_{22}F = 0$. *Explicit form of the* 8×8 *matrix* P_{ij} *is given in [17].*

2. The singular locus of the Pfaffian system (the denominator of the matrices P_{ij}*) is*

$$x_{11}^4 x_{12} x_{22}^4 (x_{12}^2 - x_{22}x_{11})^4 (x_{12}^2 + x_{22}x_{11}). \tag{69}$$

8.5.2 Deriving Hermite expansion by Difference HGM

Let $\sigma(u) = Y(u)\sin(u)$ be the ReSin function. It satisfies $u^2(\partial_u^2 + 1) \bullet \sigma(u) = 0$ as a distribution. Applying an algorithm to find annihilating ideal for a product of distributions (see, e.g., [16]), we find an annihilating operator for $Y(u)\sin(u) \cdot He_n(u)\exp(-u^2/2)$. It is also annihilated by the difference operator $S_n^2 - 2uS_n + 2(n+1)$ where S_n is the shift operator for the variable n. For example, we have $S_n \bullet He_n = He_{n+1}$. Applying an algorithm to find linear difference equation for c_n we obtain a difference operator annihilating c_n as

$$s_n^6 + 2(n+3))s_n^4 + (n^2 + 5n + 7))s_n^2 + (n+1)(n+2).$$

Initial values for the difference equation are $c_0 = \sqrt{2}F\left(\frac{1}{\sqrt{2}}\right), c_1 = \sqrt{\frac{\pi}{2e}}, c_2 = 1 - \sqrt{2}F\left(\frac{1}{\sqrt{2}}\right), c_3 = -\sqrt{\frac{\pi}{2e}}, c_4 = \sqrt{2}F\left(\frac{1}{\sqrt{2}}\right) - 2, c_5 = \sqrt{\frac{\pi}{2e}}$ where F is the Dawson's function [4] and e is Euler's number (Napier's number). These initial values are expressed in terms of special values of Dawson's integral. Here is a session by Mathematica.

[4] `https://en.wikipedia.org/wiki/Dawson_function`

```
Mathematica 11.2.0 Kernel for Linux x86 (64-bit)
Copyright 1988-2017 Wolfram Research, Inc.

In[1]:= hermiteE[n_,x_]:=2^(-n/2)*HermiteH[n,x/2^(1/2)]
In[2]:= Integrate[Sin[u]*hermiteE[0,u]*Exp[-u^2/2],{u,0,Infinity}]
                              1
Out[2]= Sqrt[2] DawsonF[-------]
                       Sqrt[2]

In[3]:= Integrate[Sin[u]*hermiteE[1,u]*Exp[-u^2/2],{u,0,Infinity}]
             Pi
Out[3]= Sqrt[---]
             2 E
```

To avoid errors in the numerical calculation of c_n by the recurrence formula, we put $d_0 = \sqrt{2}F\left(\frac{1}{\sqrt{2}}\right)$ and $d_2 = \frac{\sqrt{\pi}}{\sqrt{2e}}$ and solve the recurrence by the rational arithmetic in $\mathbf{Q}[d_1, d_2]$ and finally replace d_1, d_2 by their approximate numerical values. There are several methods to avoid such errors of difference HGM (see, e.g., [24]).

Our difference HGM gives all c_k, $k \leq 100$ in 0.009315s. On the other hand, it takes 1.6998s by Mathematica just to obtain only c_{99} on o3n.

8.6 HGM for GeLU

Let $\sigma(u)$ be the Gaussian error linear unit (GeLU) [10]

$$x(1 + \mathrm{erf}(x)) \tag{70}$$

where the error function $\mathrm{erf}(x)$ is $\frac{2}{\sqrt{\pi}}\int_0^x \exp(-t^2)dt$. Note that the GeLU of [10] is $\frac{1}{2}x(1 + \mathrm{erf}(x/\sqrt{2})$ which agrees with ours by changing the variable $x/\sqrt{2}$ to x and multiplying a scalar.

8.6.1 Expectation for GeLU

In this section, we will evaluate the expectations by the HGM. In other words, we will numerically evaluate the integral (13) by the HGM. Since we have explained how to apply the framework of the HGM to the evaluation for ReLU in Section 6, we only explain only the differences.

Proposition 2.

The GeLU $\sigma(u)$ is annihilated by the linear ordinary differential operator

$$u^2\partial_u^2 - 2u(1-u^2)\partial_u + 2(1-u^2) \tag{71}$$

Since GeLU $\sigma(u)$ is a holonomic function by the proposition, we can apply [12, Th 2] (Theorem 4) to obtain a holonomic system satisfied by $g(x)$ (13).

Theorem 10. *Let $Z(x) = \pi/\sqrt{x_{11}x_{22} - x_{12}^2}$ be the normalizing constant for the normal distribution of the covariance matrix $(-2x)^{-1}$, $x = \begin{pmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{pmatrix}$ and of the average 0.*

1. *The holonomic systems satisfied by the expectation multiplied by the normalizing constant $g_1(x) = \hat{E}[\sigma(u)\sigma(v)] = Z(x)E[\sigma(u)\sigma(v)]$ ($g(x)$ of (13) for the case that σ is GeLU) and $g_2(x) = \hat{E}[\dot{\sigma}(u)\dot{\sigma}(v)]Z(x)E[\dot{\sigma}(u)\dot{\sigma}(v)]$ ($g(x)$ of (13) for the case that σ is the derivative of GeLU) are of rank 8.*
2. *The singular locus of the Pfaffian system for $\hat{E}[\sigma(u)\sigma(v)]$ with respect to*

$$S = (1, \partial_{11}, \partial_{12}, \partial_{22}, \partial_{12}\partial_{22}, \partial_{11}\partial_{12}, \partial_{11}\partial_{22}, \partial_{11}\partial_{12}\partial_{22})^T \bullet g_i$$

is the union of zeros of $d_1 = x_{22}, d_2 = x_{22} - 1, d_3 = x_{11}, d_4 = x_{11} - 1, d_5 = x_{12}^2 - x_{22}x_{11}, d_6 = x_{12}^2 - x_{22}x_{11} + x_{22}, d_7 = x_{12}^2 + (-x_{22} + 1)x_{11}, d_8 = x_{12}^2 + (-x_{22} + 1)x_{11} + x_{22} - 1, d_9 = x_{12}^4 + (-x_{22}^2 + x_{22})x_{11}^2 + (x_{22}^2 - x_{22})x_{11}$.

This theorem can be proven in an analogously way as the proof of Theorem 7 as follows.

1. Derive the holonomic system by applying [12] (Theorem 4) and Proposition 2.
2. Translate the holonomic system by applying the restriction algorithm and an algorithm to obtain a Pfaffian system from the holonomic system. Risa/Asir codes to perform them are at [17].

These are performed in 338.8s on o3n.

8.6.2 Proof of Proposition 2

The Erf function $\mathrm{erf}(u)$ satisfies $\partial_u \bullet \mathrm{erf}(u) = \frac{2}{\sqrt{\pi}} \exp(-u^2)$. Then, it is annihilated by the operator $(\partial_u + 2u)\partial_u$, which also annihilates $f_1(u) = 1 + \mathrm{erf}(u)$. Let us derive the ODE satisfied by $\sigma(u) = f_1(u)f_2(u)$. The function $f_2(u) = u$ is annihilated by $u\partial_u - 1$. We derive a linear dependent relation for σ, $\sigma' = f_1'f_2 + f_1f_2'$ and $\sigma'' = f_1''f_2 + 2f_1'f_2' + f_1f_2''$. Since f_1 satisfies the rank 2 ODE and f_2 satisfies the rank 1 ODE, we can express σ' and σ'' in terms of f_1f_2, $f_1'f_2$ by replacing f_1'' by $-2uf_1'$, f_2' by f_2/u and f_2'' by 0. In fact, we have $\sigma = f_1f_2$, $\sigma' = f_1'f_2 + f_1f_2/u$, $\sigma'' = -2uf_1'f_2 + 2f_1'f_2/u$ and these 3 functions are linearly dependent over the rational function field $\mathbf{C}(u)$. Put the coefficients of the dependency as $c_i(u)$. Then, we have

$$c_2\sigma'' + c_1\sigma' + c_0\sigma = (c_1/u + c_0)f_1f_2 + ((-2u + 2/u)c_2 + c_1)f_1'f_2 = 0.$$

Assuming f_1f_2 and $f_1'f_2$ are linearly independent, we have $c_1/u + c_0 = 0$ and $(-2u + 2/u)c_2 + c_1 = 0$. Put $c_1 = 1$. Then, $c_0 = -1/u$ and $c_2 = \frac{1}{2u - 2/u} = \frac{u}{2u^2 - 2}$. Thus, we obtain (71) by multiplying $2u(u^2 - 1)$.

8.6.3 Evaluation of the expecation of the derivative of GeLU

We retain the notation of 8.6.2. The unnormalized expecation (15) $\hat{E}[(f(u) + g(u))(f(v) + g(v))]$ is a sum of $\hat{E}[f(u)f(v)]$, $\hat{E}[f(u)g(v)]$, $\hat{E}[g(u)f(v)]$, and $\hat{E}[g(u)g(v)]$ where $f(u) = u\,\mathrm{erf}'(u)$ and $g(u) = 1 + \mathrm{erf}(u)$. Since these functions satisfy simpler ODE's, evaluation becomes faster than utilizing the holonomic system for $g_2(x)$ in Theorem 10.

Theorem 11. *1. The unnormalized expectation* $\hat{E}[f(u)f(v)]$ *is equal to*

$$\frac{4x_{12}}{2((x_{11} - 1)(x_{22} - 1) - x_{12}^2)^{3/2}}. \tag{72}$$

2. The holonomic systems satisfied by $\hat{E}[f(u)g(v)]$ *and* $\hat{E}[g(u)g(v)]$ *are of rank* 2. *The singular locus of the Pfaffian system with respect to* $(1, \partial_{12})^T \bullet \hat{E}[f(u)g(v)]$ *is the union of zeros of* x_{22}, $x_{22} - 1$, $x_{11} - 1$, $d_1 = x_{12}^2 - x_{22}x_{11} + x_{22}$, $d_2 = x_{12}^2 + (-x_{22} + 1)x_{11} + x_{22} - 1$. *The singular locus of the Pfaffian system with respect to* $(1, \partial_{12})^T \bullet \hat{E}[g(u)g(v)]$ *is the union of zeros of* d_1, d_2, $d_3 = x_{12}^2 - x_{22}x_{11}$, $d_4 = x_{12}^2 + (-x_{22} + 1)x_{11}$, $d_5 = x_{12}^4 + (-x_{22}^2 + x_{22})x_{11}^2 + (x_{22}^2 - x_{22})x_{11}$. *These Pfaffian systems are given at [17].*

3. Values of $\hat{E}[f(u)g(v)]$ *and* $\hat{E}[g(u)g(v)]$ *at* $x = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ *are* 0 *and* $\frac{1}{2}$ *respectively. Values of* $\partial_{12} \bullet \hat{E}[f(u)g(v)]$ *and* $\partial_{12} \bullet \hat{E}[g(u)g(v)]$ *at* $x = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ *are* π *and* 1 *respectively.*

Proof. 1. $\hat{E}[f(u)f(v)]$ is a moment of the normal distribution and is easy to obtain an explicit form.
2. The Pfaffian systems are obtained in an analogous way as the proof of Theorem 7.
3. The integrals are products of single integrals for the special value of x in the statement. We can obtain an explicit form of these single integrals, e.g., with a help of Mathematica. □

Derivation of Pfaffian systems is done in 10.066s on o3n.

8.7 Degenerated Normal Distribution

When $\det(\Sigma) = 0$, the HGM for the double integral (13) cannot be applied because $\det(X) = 1/\det(-\Sigma/2)$ becomes infinity. Following the discussion of [2, p.30], we will derive a single integral representation for the expectation $E[\sigma(u)\sigma(v)]$.

Let Σ be the covariance matrix of rank 1. Then there exists a non-sigular symmetric matrix B such that $B\Sigma B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$. Let $(c_1, c_2)^T$ be the first column vector of the matrix B^{-1}. Then, the expectation for the activator σ is expressed as

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \sigma(c_1 z)\sigma(c_2 z) \exp\left(-\frac{z^2}{2}\right) dz \tag{73}$$

by [2, p.30]. To obtain the expectation for $\dot{\sigma}$, we may replace σ by $\dot{\sigma}$ in (73). In order to evaluate the integral, we may utilize the HGM for c_1 and c_2 or an efficient numerical integrator for single integrals.

References

[1] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, Ruosong Wang, On Exact Computation with an Infinitely Wide Neural Net, `https://arxiv.org/abs/1904.11955`

[2] T.W.Anderson, An Introduction to Multivariate Statistical Analysis, 2003, John Wiley & Sons, Inc.

[3] M.A.Barkatou, T.Cluzeau, C.El Bacha, J.-A.Weil, Computing Closed Form Solutions of Integrable Connections, ISSAC '12: Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation, 43–50.
`https://www.unilim.fr/pages_perso/thomas.cluzeau/Packages/IntegrableConnections/PDS.html` Y.Cho and L.Saul, Kernel methods for deep learning, Neural Information, Processing Systems (NeurIPS), 2009.

[4] A.Daniely, R.Frostig, Y.Singer, Toward Deeper Understanding of Neural Networks: The Power of Initialization and a Dual View on Expressivity, NIPS 2016, `https://arxiv.org/abs/1602.05897`

[5] M.A.Barkatou, M.Jaroschek, S.S.Maddah, Formal solutions of completely integrable Pfaffian systems with normal crossings, Journal of Symbolic Computation 81 (2017) 41–68.

[6] The analytic continuation of generalized functions with respect to a parameter, Functional analysis and its applications 6 (1972), 273–285.

[7] Y.Cho and L.Saul, Kernel methods for deep learning. In Neural Information Processing Systems (NeurIPS), 2009.

[8] I.Han, A.Zandieh, J.Lee, R.Novak, L.Xiao, A.Karbasi, Fast Neural Kernel Embeddings for General Activations, arxiv:220904121.

[9] T.Hibi et al, Gröbner Bases ; Statistics and Software systems, 2013, Springer.

[10] D.Hendrycks, K.Gimpel, Gaussian Error Linear Units (GELUs), 2016, arXiv:1606.08415.

[11] A.Jacot, F.Gabriel, C.Honger, Neural Tangent Kernel: Convergence and Generalization in Neural Networks, arxiv:1806.07572.

[12] T.Koyama, A.Takemura, Calculation of orthant probabilities by the holonomic gradient method.

[13] C.Koutschan, A Fast Approach to Creative Telescoping, Mathematics in Computer Science 4(2-3) (2010), 259-266.

[14] S.J.Matsubara-Heo, Laplace, Residue, and Euler integral representations of GKZ hypergeometric functions, arxiv:1801.04075, (2018).

[15] T.Oaku, Algorithms for b-functions, restrictions, and algebraic local cohomology groups of D-modules, Advances in Applied Mathematics 19 (1997), 61–105.

[16] T.Oaku, Y.Shiraki, N.Takayama, Algebraic algorithms for D-modules and numerical analysis, Lecture notes series on computing, computer mathematics (2003) 23–39.

[17] `http://www.math.kobe-u.ac.jp/OpenXM/Math/hgm-ntk-01`

[18] M.Petkovsek, H.S.Wilf, D.Zeilberger, A=B, AK Peters/CRC Press, 1996.

[19] Computer algebra system Risa/Asir,
`https://github.com/openxm-org/OpenXM`

[20] H.Nakayama, K.Nishiyama, M.Noro, K.Ohara, T.Sei, N.Takayama, A.Takemura, Holonomic Gradient Descent and its Application to Fisher-Bingham Integral, Advances in Applied Mathematics 47 (2011), 639–658

[21] H.Hashiguchi, Y.Numata, N.Takayama, A.Takemura, Holonomic gradient method for the distribution function of the largest root of a Wishart matrix, Journal of Multivariate Analysis, 117, (2013) 296-312,

[22] M.Saito, B.Sturmfels, N.Takayama, Gröbner Deformations of Hypergeometric Differential Equations, Algorithms and Computation in Mathematics 6, 1999, Springer.

[23] N.Takayama, T.Yaguchi, Y.Zhang, Comparison of Numerical Solvers for Differential Equations for Holonomic Gradient Method in Statistics, arxiv:2111.10947

[24] Y.Tachibana, Y.Goto, T.Koyama, N.Takayama, Holonomic gradient method for two-way contingency tables, Algebraic statistics 11 (2020) 125–153.

[25] R.Tsuchida, T.Pearce, C. van der Heide, F.Roosta, and M.Gallagher. Avoiding Kernel Fixed Points: Computing with ELU and GELU Infinite Networks. Conference on Artificial Intelligence (AAAI), 2021.

[26] `http://www.math.kobe-u.ac.jp/OpenXM/Math/hgm/ref-hgm.html`

[27] G.Yang, Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. arxiv:1902.04760.

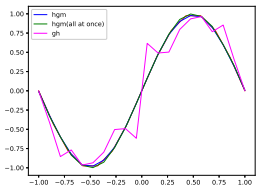

Figure 1: Inference by ReLU

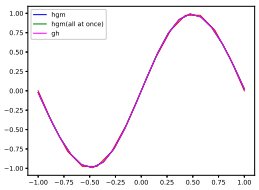

Figure 2: Inference by GeLU

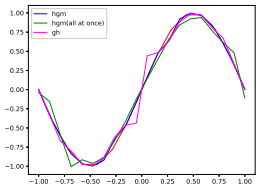

Figure 3: Inference by ReSin

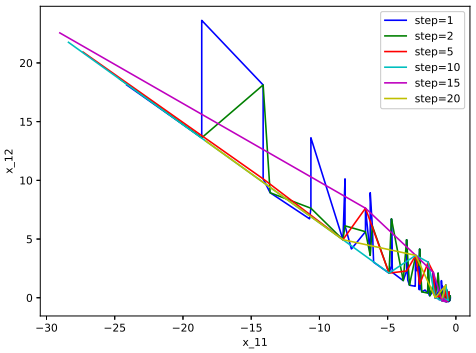

Figure 4: Integration pathes of an ODE solver.

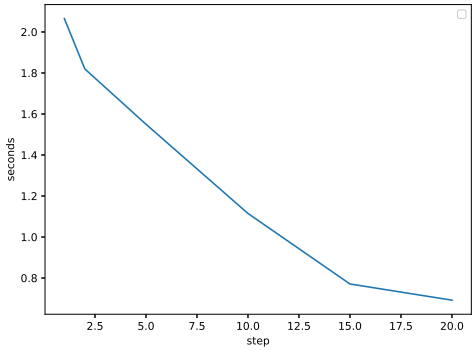

Figure 5: Timing when the parameter "step" increases.