Journal of Scientific Computing and Machine Learning, vol. 1, pp. 16–21, 2025
Submission Date: January 3, 2025 Acceptance Date: January 26, 2025

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

Akihiro Sakoda[1] **and Nobuki Takayama**[1][2]
[1,2]Department of Mathematics, Kobe University, 657-8501, Japan

Abstract

A holonomic system of linear partial differential equations is, roughly speaking, a system whose solution space is finite dimensional. A distribution that is a solution of a holonomic system is called a holonomic distribution. We give a method to numerically evaluate dual activations of holonomic activation distributions for neural tangent kernels. The method is based on computer algebra algorithms for rings of differential operators.

Keywords dual activation · neural tangent kernel · holonomic gradient method

1. Introduction

A.Jacot et al [7] introduced a kernel function $\Theta(x, x')$ that converges to the neural tanget kernel (NTK). Here, x, x' are data vectors. They show that large width limits of neural networks (NN's) can be described in terms of NTK's and Θ's. In order to construct this kernel function, we need to evalute double integrals of a function expressed in terms of an activation (distribution) and an exponential function with parameters. See (7). These double integrals are called *dual activations*. Attempts have been made to calculate dual activations for various activations, and closed formulas have been found for many activations. In particular, Han et al [5] gives several new closed formulas as well as some approximation methods based on Hermite polynomials, which work nicely for smooth activation functions.

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. A typical example of holonomic distributions is the rectified linear unit (ReLU) $\max(u, 0)$. In this paper, we note that when the activation is a holonomic distribution, its dual activation satisfies a holonomic system of linear partial differential equations [4] and further show that the holonomic system can be derived by computer algebraic algorithms. The holonomic system can be translated into a system of ordinary differential equations (ODE's) on a given smooth curve on the parameter space. We evalute numerically the dual activation by solving this system of ODE's. Accuracy and efficiency of numerical evaluations of dual activations depend on numerical solver of ODE's. Our holonomic method is particularly useful when a closed formula of a dual activation of a non-smooth holonomic activation is not known.

The method of deriving a holonomic system of definite integrals with parameters and performing a numerical analysis of it is called the holonomic gradient method (HGM) and has been applied to several problems [23]. We refer to the book [6, chap 6] and papers [16], [17] as introductory documents. Although methods proposed in this paper falls into the HGM, our method is specialized to evaluating dual activations.

2. Computation of NTK

Let $f(x,\theta)$ be a NN where x is an input vector and θ is a parameter vector. The neural tangent kernel (NTK) is defined by the inner product of the gradient $\frac{\partial f(x,\theta)}{\partial\theta}$ as

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

Here $\langle\ ,\ \rangle$ is the inner product. Let f be a totally connected NN of depth L. Following Jacot et al [7] and Arora et al [1], we define the following covariance matrices and expecations.

$$c_\sigma = \left(E_{z\sim N(0,1)}[\sigma(z)^2]\right)^{-1} \tag{1}$$

$$\Sigma^{(0)}(x,x') = x^\top x' + \beta^2, \tag{2}$$

$$\Lambda^{(h)}(x,x') = \begin{pmatrix} \Sigma^{(h-1)}(x,x) & \Sigma^{(h-1)}(x,x') \\ \Sigma^{(h-1)}(x',x) & \Sigma^{(h-1)}(x',x') \end{pmatrix} \tag{3}$$

$$\Sigma^{(h)}(x,x') = c_\sigma E_{(u,v)\sim N(0,\Lambda^{(h)})}[\sigma(u)\sigma(v)] + \beta^2 \tag{4}$$

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

Here, $N(0,\Lambda)$ is the two variable normal distribution with average 0 and covariance Λ (see, e.g., [2]), $\dot\sigma$ is the derivative of activation, and β is a hyperparameter of a strength of an effect of bias parameters. We call the expectation in (4) the *dual activation* of σ. It is shown in [7] and more precisely in [1] that the NTK converges in probability to

$$\Theta(x,x') = \sum_{h=1}^{L+1} \left(\Sigma^{(h-1)}(x,x') \prod_{h'=h}^{L+1} \dot\Sigma^{(h')}(x,x') \right). \tag{6}$$

when the width of the NN goes to infinity.

3. Holonomic activation and HGM

Let $\sigma(u)$ be an activation. When it is a solution of a linear ODE with polyonomial coefficients, we call it holonomic activation (distribution). For example, ReLU satisfies $(u\partial_u - 1) \bullet \sigma(u) = 0$ and then it is holonomic activation. Suppose that a holonomic activation σ is an analytic function. Then, it has only finite number of poles or branch points on the complex plane. Thus, for example, the sigmoid function $\frac{1}{1+e^{-x}}$ is not holonomic, because $x = (1+2k)\pi\sqrt{-1}$, $k \in \mathbf{Z}$ are poles of this function.

For a holonomic activation σ, we put

$$g(x) = \int_{\mathbf{R}^2} \sigma(u)\sigma(v)\exp(x_{11}u^2 + 2x_{12}uv + x_{22}v^2)dudv. \tag{7}$$

When we need to specify the activation σ, we denote g by $\hat{E}[\sigma(u)\sigma(v)]$. $E_{(u,v)\sim N(0,\Lambda)}[\sigma(u)\sigma(v)]$ is $g(x)\frac{\sqrt{\det(x)}}{\pi}$ where $x = (x_{ij}) = -\frac{1}{2}\Lambda^{-1}$.

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., [6, 6.4, 6.8] and [18] 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 [4] is the theoretical foundation of our method.

Theorem 1. *[4], see also, e.g., [6, 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}

$$(I + \partial_n D_n) \cap D_{n-1} \tag{8}$$

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* (8) is given by T.Oaku [11] (see also, e.g., [6, Chap 6]). If we apply a Laplace transformation $x_n \mapsto -\partial_n$, $\partial_n \mapsto x_n$, we can construct $(I + x_n D_n) \cap D_{n-1}$, which is called the *restriction ideal*, by the algorithm.

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 holomorphic 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 $\partial_i s_j = \sum_{k=1}^{r} p^i_{jk}(x) s_k$ in $R_n/(R_n I)$. The rational functions p^i_{jk} can be obtained by a Gröbner basis computation (see, e.g., [6, 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

$$\frac{\partial F}{\partial x_i} = P_i F \tag{9}$$

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

$$dF = (P_1 dx_1 + \cdots + P_n dx_n) F. \tag{10}$$

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., [6, §6.2].

Algorithm 1. (HGM)
Input: Linear ODE's ℓ_1 and ℓ_2 annihilating $\sigma_1(u)$ and $\sigma_2(u)$ respectively. A curve on the x space.
Output: Values[2] of $\hat{E}[\sigma_1(u)\sigma_2(v)]$ on the curve.

1. *Apply [8, Th 2] 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 $D_5 = \mathbf{C}\langle x_{11}, x_{12}, x_{22}, y_1, y_2, \partial_{11}, \partial_{12}, \partial_{22}, \partial_1, \partial_2 \rangle$.*

2. *Apply a restriction algorithm [11] to find generators of $I_2 := (I_1 + y_1 D_5 + y_2 D_5) \cap \mathbf{C}\langle x_{11}, x_{12}, x_{22}, \partial_{11}, \partial_{12}, \partial_{22} \rangle$.*

3. *Translate I_2 into 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 the given curve.*

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

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

4. Experiments

1. We evaluate Θ for a one hidden layer NN with the activation ReLU and 1 dimensional input and output. We compare the following 4 methods.

1. The closed formula of the dual activation (e.g., [1, I]).

2. Evaluate $\hat{E}$ by the Monte-Carlo method with 5000 samples.

3. Gauss-Hermite quadrature formula of degree 10 [5, 3.3] (Gauss-Herm, gh).

4. Algorithm 1 (HGM).

Training data are values of $\sin(\pi x)$ at 15 points on $(-1, 1)$ with same distance. Test inference inputs are 20 points on $(-1, 1)$ with same distance. We performed ridge regression with a regularization parameter $\lambda = 0.001$. Steps 1, 2, 3 of Algorithm 1 can be done in 0.196s on the Risa/Asir computer algebra system [15]. We use `solve_ivp` of scipy with `rtol = 1e-10`, `atol = 1e-10` as an ODE solver. The following timing data is taken on AMD EPYC 7552 48-core[3] processor of 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

[2] (7) is the case of $\sigma = \sigma_1 = \sigma_2$.
[3] We use only 1 core.

	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

The matrix $(\Theta(x_i, x_j))$ is called the *Gram matrix* where x_i and x_j are input data. The kernel error is the mean squared error (MSE) of the Gram matrix of a test method and that evaluated by the closed formula. The "pred error" is the MSE of the outputs of a test method and those by the Gram matrix evaluated by the closed formula. The HGM is about 2.6 times slower than the Gauss-Hermite formula, but the kernel error and the prediction error are smaller. In fact, the HGM gives the exact shape of the sin curve. See Figure 1.

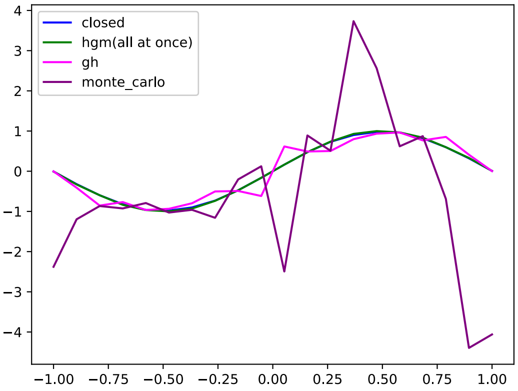

Figure 1: Learning a sin curve.

2. We evaluate Θ for a one hidden layer NN with the activation ReLU and 784 dimensional input and 2 dimensional output. Training data are 100 pictures of hand written numbers 0 and 1 of MNIST. Test inference data are 20 pictures. Timing data is taken on a machine with Intel(R) Xeon(R) Gold 6426Y (800MHz) and NVIDIA A800 40GB. The GPU is used only for computing inner products of vectors. In the timing and error table below, the Gram matrices are for 120 input points.

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

The correct answer rate is 100 % except the Monte-Carlo method, whose rate is 85 %. The kernel error of Monte-Carlo method is smaller than that of Gauss-Herm, but the Monte-Carlo method sometimes gives wrong values of $\Theta(x, x')$ which cause a recognition error.

These experiments demonstrate that the HGM will be useful to evaluate dual activations for non-smooth holonomic activations. Source codes for these experiments are obtainable from [13]. Experiments for some other activations are presented in [19].

5. Closed formula for rectified monomials of two variables

The Gauss-Hermite quadrature and the HGM are useful methods when closed formulas of dual activations are not known. However, they are slower and less accurate than closed formulas. Then, it will be important to make efforts to find new closed formulas.

Applying the HGM algorithm by hand, we obtain the following new closed formula of dual activations.

Let m, n are non-negative integers and $Y(u)$ the Heaviside function. Assume $x_{11}, x_{22} < 0$ and $0 \leq \frac{x_{12}^2}{x_{11}x_{22}} < 1$. Put

$$\varphi_1 \quad := \quad (-x_{11})^{-\alpha}(-x_{22})^{-\beta}{}_2F_1\left(\alpha, \beta, \frac{1}{2}; z\right) \tag{12}$$

$$\begin{aligned}\varphi_2 \quad := \quad & (-x_{11})^{-\alpha}(-x_{22})^{-\beta}\sqrt{z}\,\mathrm{sign}\,(x_{12}) \\ & \times {}_2F_1\left(\alpha+\frac{1}{2}, \beta+\frac{1}{2}, \frac{3}{2}; z\right)\end{aligned} \tag{13}$$

where ${}_2F_1$ is the Gauss hypergeometric function,

$$\alpha = \frac{1+m}{2}, \beta = \frac{1+n}{2}, z = \frac{x_{12}^2}{x_{11}x_{22}} \tag{14}$$

and $\mathrm{sign}\,(x)$ is the sign of x. We denote by $\hat{E}$ the unnormalized expectation $g(x)$ (7).

Theorem 2. *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*[4] *to*

$$\frac{1}{4}\Gamma(\alpha)\Gamma(\beta)\varphi_1 + \frac{1}{2}\Gamma\left(\alpha+\frac{1}{2}\right)\Gamma\left(\beta+\frac{1}{2}\right)\varphi_2 \quad (15)$$

A proof of this theorem is given in [19][5]. Note that we have

$$ {}_2F_1((1+m)/2, 1/2, 1/2; z) = (1-z)^{-1/2-m/2}, \quad (16)$$

$$ {}_2F_1(1,1,1/2;z) = \left(1 + \frac{\sqrt{z}\arcsin(\sqrt{z})}{\sqrt{1-z}}\right)(1-z)^{-1}, \quad (17)$$

$$ {}_2F_1(3/2,3/2,3/2;z) = (1-z)^{-3/2}. \quad (18)$$

A closed formula of the dual activation of a rectified monomial of one variable is given in [3]. A closed formula of the dual activation of a polynomial is given in [5, Th 1] by utilizing Hermite polynomials. Since we have, for constants c_i,

$$\begin{aligned} & \hat{E}[(c_1u^{m_1} + c_2u^{m_2})Y(u)(c_1v^{m_1} + c_2v^{m_2})Y(v)] \\ = & \sum_{i,j=1}^{n} c_ic_j\hat{E}[u^{m_i}v^{m_j}Y(u)Y(v)], \end{aligned} \quad (19)$$

our theorem gives a closed formula of the dual activation of a rectified polynomial in terms of the Gauss hypergeometric functions with degenerated parameters.

Acknowledgments

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

References

[1] S.Arora, S.S.Du, W.Hu, Z.Li, R.Salakhutdinov, R.Wang, On Exact Computation with an Infinitely Wide Neural Net, https://arxiv.org/abs/1904.11955, NeurIPS 2019.

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

[3] Y.Cho, L.Saul, Kernel Methods for Deep Learning, NeurIPS 2009.

[4] I.N.Bernstein, The Analytic Continuation of Generalized Functions with respect to a Parameter, Functional analysis and its applications 6 (1972), 273–285.

[5] I.Han, A.Zandieh, J.Lee, R.Novak, L.Xiao, A.Karbasi, Fast Neural Kernel Embeddings for General Activations, https://arxiv.org/abs/220904121, NeurIPS 2022.

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

[7] A.Jacot, F.Gabriel, C.Honger, Neural Tangent Kernel: Convergence and Generalization in Neural Networks, https://arxiv.org/abs/1806.07572, NeurIPS 2018.

[8] T.Koyama, A.Takemura, Calculation of Orthant Probabilities by the Holonomic Gradient Method, Japan Journal of Industrial and Applied Mathematics 32 (2015), 187–204.

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

[10] S.J.Matsubara-Heo, Laplace, Residue, and Euler Integral Representations of GKZ Hypergeometric Functions, https://arxiv.org/abs/1801.04075.

[11] T.Oaku, Algorithms for b-functions, Restrictions, and Algebraic Local Cohomology Groups of D-modules, Advances in Applied Mathematics 19 (1997), 61–105.

[12] T.Oaku, Y.Shiraki, N.Takayama, Algebraic Algorithms for D-modules and Numerical Analysis, Lecture Notes Series on Computing, Computer Mathematics (2003), 23–39.

[13] https://www.math.kobe-u.ac.jp/OpenXM/Math/hgm-ntk-02

[4] [10, §3] gives a Laplace type integral representation (whose integral kernel contains the exponential function) of ${}_2F_1$ as a speical case.

[5] This paper is an extended abstract of [19].

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

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

[16] 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

[17] 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,

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

[19] A.Sakoda, N.Takayama, An Application of the Holonomic Gradient Method to the Neural Tangent Kernel, https://arxiv.org/abs/2410.23626.

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

[21] Y.Tachibana, Y.Goto, T.Koyama, N.Takayama, Holonomic Gradient Method for Two-way Contingency Tables, Algebraic statistics 11 (2020) 125–153.

[22] 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.

[23] https://www.math.kobe-u.ac.jp/OpenXM/Math/hgm/ref-hgm.html