# Orthogonal Polynomials Quadrature Algorithm: a functional analytic approach to inverse problems in deep learning


**Anonymous Author(s)**
Affiliation
Address
email



## Abstract

We present the new Orthogonal Polynomials–Quadrature Algorithm (OPQA), a parallelizable algorithm that solves two common inverse problems in deep learning from a functional analytic approach. First, it finds a smooth probability density function as an estimate of the posterior, which can act as a proxy for fast inference; second, it estimates the evidence, which is the likelihood that a particular set of observations can be obtained. Everything can be parallelized and completed in one pass.

A core component of OPQA is a functional transform of the square root of the joint distribution into a special functional space of our construct. Through this transform, the evidence is equated with the $L^2$ norm of the transformed function, squared. Hence, the evidence can be estimated by the sum of squares of the transform coefficients.

To expedite the computation of the transform coefficients, OPQA proposes a new computational scheme leveraging Gauss–Hermite quadrature in higher dimensions. Not only does it avoid the potential high variance problem associated with random sampling methods, it also enables one to speed up the computation by parallelization, and significantly reduces the complexity by a vector decomposition.


## 1 Introduction

Let $P$ be a probability density function, $X = (x_i)_{i=1}^D$ be a set of observations and $\theta = (\theta_i)_{i=1}^N$ be the set of (unknown) latent variables. We are interested in the posterior

$$P(\theta|X) := \frac{P(\theta, X)}{\int_\theta P(\theta, X)} \tag{1}$$

and the evidence

$$P(X) = \int_\theta P(\theta, X). \tag{2}$$

In most cases, there are limitations that make it impractical to compute the posterior or the evidence directly (see Appendix 3.3). For posterior inference, there are two major approaches. The first approach is random sampling, including Markov chain Monte Carlo methods such as the Metropolis–Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) and the Hamilton Monte Carlo algorithm (Hoffman & Gelman, 2014). The second approach is the proxy model approach, including variational inference which was first developed about three decades ago (Peterson & Anderson, 1987; Hinton & Camp, 1993; Waterhouse et al., 1996; Jordan et al., 1999). The idea behind variational inference is to find the optimal proxy of the posterior by means of optimization.


Submitted to DLDE-III Workshop (NeurIPS 2023). Do not distribute.

In this paper, we introduce a new approximation approach, the **Orthogonal Polynomials–Quadrature Algorithm (OPQA)** (see *Problem Statement* in Section 2.1). Polynomials have been a staple tool in the world of mathematical physics (Simon, 1971; Vinck et al., 2012), approximation theory (Deift, 2000) and statistics (Walter, 1977; Diaconis et al., 2008). However, applications of orthogonal polynomials to machine learning are scarce to the best of our knowledge.

It is also important to note that even though both OPQA and Polynomial Chaos Expansion (PCE) involve the use of polynomials, they are completely different in nature. The most crucial difference is that the orthogonality of the basis of OPQA is with respect to the measure $d\nu_{1:N}$ of our construct in equation (5), while for PCE the orthogonality is with respect to a known prior. For most of the problems we study, the prior is not known at all.

# 2 OPQA: Problem Statement, Algorithm and Computation Scheme

## 2.1 Problem Statement

OPQA accomplishes two goals: first, it expresses the evidence as a series

$$P(X) = \sum_{\tau} |a_\tau|^2, \tag{3}$$

where $a_\tau$ are the coefficients of $P(\theta, X)$ of a special functional transform of our choice (see eq. (11)). This expression allows one to attain the second goal, which is to get a smooth estimate of the posterior, $P(\theta|X)$ by a probability density function $f_T(\theta)$, that is,

$$P(\theta|X) \approx f_T(\theta) := p_T(\theta)^2 \prod_{j=1}^{N} e^{-\theta_j^2} \geq 0, \tag{4}$$

where $p_T(\theta)$ is a multivariate polynomial and $\int_{\mathbb{R}^N} f_T(\theta) d\theta = 1$.

## 2.2 Outline of the Algorithm

We consider the functional space $L^2(d\nu_{1:N})$ associated with the following measure on $\mathbb{R}^N$

$$d\nu_{1:N}(\theta) := \prod_{j=1}^{N} e^{-\theta_j^2} d\theta_j, \tag{5}$$

where $\theta = (\theta_1, \theta_2, \cdots, \theta_N) \in \mathbb{R}^N$. Let $h_i(x)_{i=0}^{\infty}$ be the normalized one-dimensional Hermite polynomials that are orthogonal with respect to the measure $d\nu = e^{-\theta^2} d\theta$ on $\mathbb{R}$, that is,

$$\int_{\mathbb{R}} h_i(x) h_j(x) e^{-\theta^2} dx = \delta_{ij} \tag{6}$$

Such orthogonality implies that the tensor products of Hermite polynomials of the form

$$\phi_\tau(\theta) := h_{i_1}(\theta_1) h_{i_2}(\theta_2) \cdots h_{i_N}(\theta_N) \tag{7}$$

form an orthogonal polynomial basis that is orthogonal with respect to $d\nu_{1:N}$. The $N$-tuple $\tau = (i_1, i_2, \ldots, i_N)$ is known as a **multi-index**.

The measure $d\nu_{1:N}$ is special because it fulfills the finite moment criterion (22). Hence, by the Riesz Theorem (Theorem 3.1), the family of polynomials is dense in $L^2(d\nu_{1:N})$. In particular, observe that the square root of the following function is in $L^2(d\nu_{1:N})$,

$$\tilde{P}(\theta, X) := P(\theta, X) \prod_{j=1}^{N} e^{\theta_j^2} \tag{8}$$

and its $L^2$ norm squared is the evidence:

$$\|\tilde{P}^{1/2}\|^2_{L^2(d\nu_{1:N})} = \left( \int_{\mathbb{R}^N} \left| \tilde{P}(\theta, X)^{1/2} \right|^2 d\nu_{1:N} \right) = \left( \int_{\mathbb{R}^N} |P(\theta, X)| d\theta_{1:N} \right) = P(X), \tag{9}$$

which is finite. Next, we transform $\tilde{P}(\theta, X)^{1/2}$ into an infinite series by projecting it onto the polynomial basis $(\phi_\tau)_\tau$. The transform coefficients are given by

$$a_\tau = \left\langle \tilde{P}(\theta, X)^{1/2}, \phi_\tau \right\rangle_{d\nu_{1:N}}, \tag{10}$$

which is equivalent to

$$a_\tau := \int_{\mathbb{R}^N} P(\theta, X)^{1/2} \phi_\tau(\theta) \left( \prod_{j=1}^{N} e^{-\theta_j^2/2} \right) d\theta_{1:N}. \tag{11}$$

Recall that this polynomial basis is dense due to Riesz' Theorem. Suce density allows one to invoke the Parseval Identity, which equates the $L^2$-norm with the sum of its transform coefficieints, that is,

$$\|\tilde{P}^{1/2}\|^2_{L^2(d\nu_{1:N})} = \sum_\tau a_\tau^2. \tag{12}$$

Combining this with (9), we obtain one of our two results,

$$P(X) = \sum_\tau a_\tau^2. \tag{13}$$

The fact that the coefficients $(a_\tau)_\tau$ are absolutely convergent implies that the summation can be executed in any order. Furthermore,

$$\tilde{P}(\theta, X)^{1/2} \approx \sum_\tau a_\tau \phi_\tau(\theta), \tag{14}$$

which id equivalent to

$$P(\theta, X) \approx \left( \sum_\tau a_\tau \phi_\tau(\theta) \right)^2 \prod_{j=1}^{N} e^{-\theta_j^2} \tag{15}$$

Combining with $P(\theta|X) = P(\theta, X)/P(X)$ and $P(X) > 0$, we obtain

$$P(\theta|X) \approx p_T(\theta)^2 \prod_{j=1}^{N} e^{-\theta_j^2}, \tag{16}$$

where

$$p_T(\theta) := \left( \sum_{\tau \in T} |a_\tau|^2 \right)^{-1/2} \left( \sum_{\tau \in T} a_\tau \phi_\tau(\theta) \right). \tag{17}$$

Observe that the right hand side of 16 is a probability density function because

$$\int p_T(\theta)^2 \left( \prod_{j=1}^{N} e^{-\theta_j^2} \right) d\theta = \left( \sum_{\tau \in T} |a_\tau|^2 \right)^{-1} \sum_{\tau, \sigma \in T} \left( \int a_\tau a_\sigma \phi_\tau(\theta) \phi_\sigma(\theta) d\mu_{1:N} \right) = 1. \tag{18}$$

The last equality follows from the fact that $\phi_\tau$ are orthogonal polynomials with respect to $d\nu_{1:N}$, so the integral (inside the parenthesis) is zero for $\tau \neq \sigma$, and $a_\tau^2$ otherwise.

### 2.3 Outline of the Computation Scheme

Due to the unique nature of the measure $(\prod_{j=1}^{N} e^{-\theta_j^2/2}) d\theta_{1:N}$. in (11), we propose the use of Gauss–Hermite quadrature to estimate $a_\tau$. Not only does it expedite the computation by allowing parallelization, it reduces the high variance problems caused by random sampling methods.

The readers should be reminded that the following computational scheme could be further optimized, and has no bearing on the mathematical correctness of the algorithm.

First, we choose a quadrature order $\Gamma$. Quadrature of order $\Gamma$ works well to approximate function which can be well estimated by a polynomial of degree $2\Gamma - 1$. For that reason, usually a single-digit $\Gamma$ will suffice.

**Algorithm 1** The Orthogonal Polynomials–Quadrature Algorithm (OPQA)

**Input** Quadrature order $\Gamma$. Joint distribution $P(\theta, X)$.
**Output** Coefficients $(a_\tau)_{\tau\in T}$ which can be used to compute evidence $P(X)$ and a smooth probability density function $f_T(\theta)$ that estimates $P(\theta|X)$.
**while** $\Sigma = \sum_{|\tau|<d} |a_\tau|^2$ does not converge **do**
  Increase the degree $d$ by $1$.
  Compute $h_d(x)$ for $x = \tilde{r}_1, \ldots, \tilde{r}_\Gamma$.
  **for** multi-index $\tau$ of degree $d$ **do**
    Compute $\Phi_\tau = (\phi_\tau(\tilde{r}^{(j)}))_{(j)\in I}$
    Compute $a_\tau = \vec{\Pi} \cdot \Phi_\tau$
    Add $a_\tau^2$ to $\Sigma$.
  **end for**
**end while**

From the one-dimensional Gauss quadrature nodes and weights $(r_i, w_i)_{i=1}^{\Gamma}$[1], we form our multivariate nodes in $\mathbb{R}^N$, $(\tilde{r}^{(j)})_{(j)}$; and weights, $(\tilde{w}^{(j)})_{(j)}$ for each grid-index $(j) \in I$. The transform coefficients can then be estimated by

$$a_\tau \approx \sum_{(j)\in I} \tilde{w}_{(j)} P(\tilde{r}^{(j)}, X)^{1/2} \phi_\tau(\tilde{r}^{(j)}). \tag{19}$$

The right hand side of (19) can be expressed by vectors $\vec{W}$, $\vec{P}$ and $\vec{\Phi}_\tau$ as[2]

$$a_\tau \approx \vec{W} \odot \vec{P} \cdot \vec{\Phi}_\tau. \tag{20}$$

This decomposition into three vectors will bring many computational advantages that help tackle the problem of dimensionality, with the major advantages being: (1) most of the values can be obtained from simple arithmetic, and (2) both $\vec{W}$ and $\vec{\Phi}_\tau$ depend on values of size $O(\Gamma \cdot d)$ where $d$ is the degree of polynomial estimation; and (3) the expression (20) allows parallelization, which substantially increases the speed of computation.

### 2.4 Our Contributions

**A new functional analytic perspective.** Instead of finding a proxy through optimization, we identified a special functional transform onto $L^2(d\nu_{1:N})$ (see equation (8)) such that the evidence $P(X)$ is equal to the sums of squares of the transform coefficients.

**Leveraging the density of polynomials and Parseval Identity.** The measure $\nu_{1:N}$ is special because it satisfies the moment criteria of the Riesz Theorem, which ensures the density of polynomials in $L^2(d\nu_{1:N})$, which ensure that (13) is an equality.

**A flexible and scalable approach.** OPQA does not require any assumptions about the prior or the independence of the latent variables. Furthermore, OPQA can produce arbitrarily good approximations as we increase the degree of polynomial approximation and order of quadrature $\Gamma$.

**An accurate, parallelizable and efficient computation scheme.** By using quadrature, it counters the variance problems from random sampling methods; the discretization of $a_\tau$ in (20) allows for efficient computation. In particular, both $\vec{W}$ and $\vec{\Phi}_\tau$ are independent of the distribution in question $P$, so both $\vec{W}$ and $\vec{\Phi}_\tau$ are essentially universal constants that apply to all OPQA applications. Besides, $\vec{W}$ only depends on $\Gamma$ quadrature weights; and $\vec{\Phi}_\tau$ on a set of $\Gamma \cdot |\tau|$ values, namely,

$$V_{|\tau|} := \{h_d(\tilde{r}_i) : 0 \leq d \leq |\tau|; 1 \leq i \leq \Gamma\}. \tag{21}$$

[1] These constants are available in Numpy libraries and numerical analysis handbooks such as Abramowitz & Stegun (1972).

[2] The symbols $\odot$ and $\cdot$ denote the pointwise multiplication and dot product of two vectors respectively.

# 3 Appendix

## 3.1 Appendix: Supporting Theorems

In Section 2.2, it was shown that OPQA relies on the Riesz Theorem, which guarantees the density of polynomials in $L^2(d\nu_{1:N})$ if the measure $d\nu_{1:N}$ satisfies the moment condition (22), which is a classic result in approximation theory.

**Theorem 3.1** (Density of polynomials in $L^2$)**.** *(Riesz, 1922). Let $\nu$ be a measure on $\mathbb{R}^N$ satisfying*

$$\int_{\mathbb{R}^N} e^{c|\theta|} d\nu < \infty \tag{22}$$

*for some constant $c > 0$, where $|\theta| = \sum_{j=1}^N |\theta_j|$; then the family of polynomials is dense in $L^2(\nu)$. In other words, given any $f \in L^2(\nu)$, there is a sequence of polynomials $f_n(\theta)$ such that*

$$\lim_{n\to\infty} \int_{\mathbb{R}^N} |f(\theta) - f_n(\theta)|^2 d\nu = 0. \tag{23}$$

Related moment problems are discussed in depth by Akhiezer (1965) (Theorem 2.3.3 and Corollary 2.3.3). A nice short proof of the result was presented in Schmuland (1992).

An important implication of criterion (22) is that all polynomials are in $L^p(\nu)$, for any $p \geq 1$. To see that, it suffices to show that for any $c > 0$ and integer $k \geq 0$

$$\lim_{x\to\infty} \frac{x^k}{e^{cx}} < \infty \tag{24}$$

via the repeated application of the L'Hôpital rule.

## 3.2 Appendix: Hermite Polynomials and Density of Polynomials

Hermite polynomials[3] are polynomials on $\mathbb{R}$ that are orthogonal with respect to the measure

$$d\nu := e^{-x^2} dx \text{ on } \mathbb{R}. \tag{25}$$

Hermite polynomials satisfy the following orthogonality relation

$$\int_{\mathbb{R}} H_m(x) H_n(x) d\nu(x) = \sqrt{\pi} 2^n n! \delta_{nm}. \tag{26}$$

Normalized Hermite polynomials are denoted as $h_n(x) := H_n(x)/\|H_n\|$. The Hermite polynomials used in this paper are

$$H_0(x) = 1, \qquad h_0(x) = \pi^{-1/4}$$
$$H_1(x) = 2x, \quad h_1(x) = \sqrt{2}\pi^{-1/4} x$$

and the higher order polynomials can be obtained from the following recurrence relation

$$H_{n+1}(x) = 2x H_n(x) - 2n H_{n-1}(x). \tag{27}$$

The measure $\nu$ is the building block of the measure $\nu_{1:N}$ defined in equation (5). A critical property of $\nu$ is that it has finite moments, that is, there is a constant $c > 0$ such that

$$\int_{\mathbb{R}} e^{c|\theta|} d\nu \leq 2 \int e^{-(\theta - c/2)^2 + \frac{c^2}{4}} d\theta < \infty. \tag{28}$$

Following a similar argument, one can prove that

$$\int_{\mathbb{R}^N} e^{c|\theta|} d\nu_{1:N}(\theta) = \prod_{j=1}^N \int e^{c|\theta_j|} e^{-\theta_j^2} d\theta_j < \infty. \tag{29}$$

Condition (29) makes $\nu_{1:N}$ eligible for the Riesz Theorem (Theorem 3.1), which implies the density of polynomials in $L^2(\nu_{1:N})$. Without this density, the equality (13) may not hold.

Apart from Hermite polynomials, Chebyshev's polynomials and Jacobi polynomials are among the most commonly known families of orthogonal polynomials. For a comprehensive introduction to orthogonal polynomials, the readers may refer to Simon (2005); Koornwinder (2013).

[3] The Hermite polynomials used in this paper are often known as the physicists' Hermite polynomials because they are orthogonal to $e^{-x^2}$ instead of $e^{-x^2/2}$.

### 3.3 Appendix: Example of a Gaussian Mixture Model with 3 Clusters

We ran this experiment: first, we sampled $N_0 = 3$ points, $\mu \sim N(0, 10)$ and obtained $\mu_1 = -18.61$, $\mu_2 = 3.81$ and $\mu_3 = 8.84$. Then we generated $n = 1000$ samples by first randomly selecting an integer $i$ from $[1, 2, 3]$, and then drawing $x \sim N(\mu_i, 1)$. Figure 1 presents a plot of the joint distribution $p(x, \mu_1, \mu_2, \mu_3)$ of this particular experiment, alongside with a normalized histogram of these $1000$ samples.

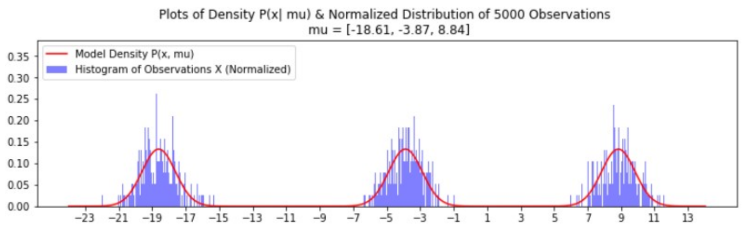


Figure 1: Example of a Mixed Gaussian Model.

In general, we are interested in the inverse problem of approximating the posterior

$$P(\mu|x_{1:n}) : \mathbb{R}^N \mapsto \mathbb{R} \tag{30}$$

as a function of the latent variables $\mu \in \mathbb{R}^N$ given the observations $x_{1:n}$. Observe that the joint probability density function is given by

$$P(\mu_{1:K_0}, x_{1:n}) = \prod_{k=1}^{K_0} p(\mu_k) \prod_{i=1}^{n} p(x_i|\mu_{1:K_0}). \tag{31}$$

To obtain the posterior in (30), one needs the normalizing weight $P(x_{1:n})$, which requires us to sum (31) in $k$ and integrate in $\mu_{1:K_0}$ . First, note that for any one sample $x$,

$$P(x|\mu_{1:K_0}) = \sum_{k=1}^{K_0} p(x, z_k|\mu_{1:K_0}) = \sum_{k=1}^{K_0} p(\mu_k|\mu_{1:K_0}) p(x, \mu_k) = \frac{1}{K_0} \sum_{k=1}^{K_0} p(x, \mu_k). \tag{32}$$

Then we need to integrate (32) against $d\mu_{1:K_0}$. That results in the following formula

$$P(x_{1:n}) = \int_{\mu_{1:K_0}} \prod_{k=1}^{K_0} p(\mu_k) \prod_{i=1}^{n} \left( \sum_{k=1}^{K_0} \frac{1}{K_0} p(x_i, \mu_k) \right) d\mu_{1:K_0}. \tag{33}$$

While it may be possible to compute (33) directly, the computation is far from straightforward. Furthermore, there are $K_0^n$ terms, making the computations extremely expensive as $n$ increases.

To illustrate the aforementioned point, we consider the simplest case of just one latent variable $\mu$ and one sample $x$. We chose this particular example because the evidence comes in closed form and it will allow us to compute the ground truth evidence.

The evidence (33) is given by

$$P(x) = \int_{\mathbb{R}} p(\mu) p(x|\mu) d\mu = \frac{1}{2\pi\sigma_\mu\sigma_x} \int_{\mathbb{R}} e^{-\frac{\mu^2}{2\sigma_\mu^2}} e^{-\frac{(x-\mu)^2}{2\sigma_x^2}} d\mu \tag{34}$$

Expanding the function inside the integral, we get

$$(34) = \frac{\exp\left(-\frac{x^2}{2\sigma_x^2}\right)}{2\pi\sigma_\mu\sigma_x} \int_{\mathbb{R}} \exp\left(-\left(\frac{1}{2\sigma_\mu^2} + \frac{1}{2\sigma_x^2}\right)\mu^2 + \frac{x}{\sigma_x^2}\mu\right) d\mu. \tag{35}$$

We perform a change of variable

$$t = \left( \sqrt{\frac{1}{2\sigma_\mu^2} + \frac{1}{2\sigma_x^2}} \right) \mu \tag{36}$$

and let

$$C_0 := \sqrt{\frac{1}{2\sigma_\mu^2} + \frac{1}{2\sigma_x^2}}. \tag{37}$$

Then

$$(35) = \frac{\exp\left(-\frac{x^2}{2\sigma_x^2}\right)}{2C_0\pi\sigma_\mu\sigma_x} \int_{\mathbb{R}} \exp\left(-t^2 + \frac{x}{C_0\sigma_x^2} t\right) dt. \tag{38}$$

Note that

$$-t^2 + \frac{x}{C_0\sigma_x^2} = -\left(t - \frac{x}{2C_0\sigma_x^2}\right)^2 + \left(\frac{x}{2C_0\sigma_x^2}\right)^2 \tag{39}$$

and that $\int e^{-y^2} dy = \sqrt{\pi}$. Combining all of these, we arrive at the following expression for the evidence

$$P(x) = \frac{\exp\left(-\frac{x^2}{2\sigma_x^2}\right) \exp\left(\left(\frac{x}{2C_0\sigma_x^2}\right)^2\right)}{2C_0\sqrt{\pi}\sigma_\mu\sigma_x}. \tag{40}$$

Furthermore, we simplify the expressions involving $C_0$ in (defined in (37)) and we get

$$P(x) = \frac{\exp\left(-\frac{x^2}{2\sigma_x^2}\right) \exp\left(\frac{x^2\sigma_\mu^2}{2\sigma_x^2(\sigma_x^2+\sigma_\mu^2)}\right)}{\sqrt{2\pi}\sqrt{\sigma_\mu^2 + \sigma_x^2}} = \frac{\exp\left(\frac{-x^2}{2(\sigma_x^2+\sigma_\mu^2)}\right)}{\sqrt{2\pi}\sqrt{\sigma_\mu^2 + \sigma_x^2}}. \tag{41}$$

# References


Abramowitz, M. and Stegun, I. A. *Handbook of Mathematical Functions*. Dover, 1972.

Akhiezer, N. I. *The Classical Moment Problem and Some Related Questions in Analysis*. Dover Publications, 1965.

Deift, P. *Orthogonal Polynomials and Random Matrices: A Riemann-Hilbert Approach*, volume 3. Courant Lecture Notes. American Mathematical Society, 2000.

Diaconis, P., Khare, K., and Saloff-Coste, L. Gibbs sampling, exponential families and orthogonal polynomials. *Statistical Science*, 23(2):151–178, 2008. doi: https://doi.org/10.1214/07-STS252.

Hastings, W. K. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–103, 1970.

Hinton, G. and Camp, D. V. Keeping the neural networks simple by minimizing the description length of the weights. *Computational Learning Theory*, pp. 5–13, 1993.

Hoffman, M. D. and Gelman, A. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15:1593–1623, 2014.

Jordan, M. I., Ghahramani, Z., Jaakkola, T., and Saul, L. Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

Koornwinder, T. Orthogonal polynomials, a short introduction. In *C. Schneider, J. Bluemlein J. (eds) Computer Algebra in Quantum Field Theory. Texts & Monographs in Symbolic Computation (A Series of the Research Institute for Symbolic Computation, Johannes Kepler University, Linz, Austria)*, pp. 145–170. Springer, Vienna, 2013.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., and Teller, A. H. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21, 1953.

Peterson, C. and Anderson, J. R. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1(5):995–1019, 1987.

Riesz, M. Sur le problème des moments et le théorème de parseval correspondant. *Acta Sci. Math. Szeged*, 1:209–225, 1922.

Schmuland, M. Dirichlet forms with polynomial domain. *Math. Japonica*, 37(6):1015–1024, 1992.

Simon, B. Distributions and their hermite expansions. *Journal of Mathematical Physics*, 12(1), 1971.

Simon, B. *Orthogonal polynomials on the unit circle. Part 1 & Part 2*, volume 54. American Mathematical Society, 2005.

Vinck, M., Battaglia, F. P., Balakirsky, V. B., Vinck, A. J. H., and Pennartz, C. M. A. Estimation of the entropy based on its polynomial representation. *Phys. Rev. E*, 85(5), 2012. doi: https://doi.org/10.1103/PhysRevE.85.051139.

Walter, G. G. Properties of hermite series estimation of probability density. *Annals of Statistics*, 5(6):1258–1264, 1977.

Waterhouse, S., MacKay, D., and Robinson, T. Bayesian methods for mixtures of experts. *Neural Information Processing Systems*, 1996.