Infinitely wide Transformer networks and their Laplace operators

Interdisciplinary Projects for Informatics: SS22

Hanady Gebran
Advisor: Dr. rer. nat. Felix Dietrich
Supervisor: Prof. Dr. Hans-Joachim Bungartz
Chair: Scientific Computing in Computer Science

October 23, 2022

Contents

1 Abstract

The most commonly used neural network architectures with i.i.d. prior on their parameters are equivalent in the limit of infinite network width, to a Gaussian process (GP). This correspondence allows exact Bayesian inference by evaluating the corresponding GP. The GP kernel of an infinite-width network can be studied via the geometry it induces on the data. This induction is possible because all NNGP kernels are symmetric positive definite kernels and thus can be used to span function spaces on the data manifold on which the original network was trained. We seek to establish this association through the relationship between the integral operator of the kernels associated with networks and the Laplace-Beltrami operator on manifolds. Indeed, the Laplace-Beltrami operator is a core element whose eigenfunctions have a multiscale structure related to spatial discretization schemes such as sparse grids and can retrieve the geometry of a Riemannian manifold. We attempted to establish a direct connection between the neural network kernels and the family of heat kernels which in turn can generate under some conditions the Laplace-Beltrami operator, but it was unsuccessful. We also studied the eigenvalues and then the orthogonal family of the integral operator of the kernel corresponding to a neural network to find a transformation that could map this family to the basis generated by the Laplace-Beltrami operator and thus define a construction: this work is still in process because the mapping is not quite straightforward. And lastly, we studied one of the actual state-of-the-art approaches that characterize function sets of Reproducing kernel Hilbert space of Neural Tangent Kernel for a few neural network architectures. For the application, we consider an NLP field by studying transformers. We deployed the NNGP and NTK transformer kernels on an IMDb movie dataset for sentiment analysis.

2 Acknowledgements

I am deeply indebted to my advisor, Dr. Felix Dietrich, for his constant guidance during the IDP, for his clarification of areas that were completely unknown to me before this research, and for his many suggestions, ideas, and constructive criticism. I would also like to express my gratitude to my supervisor, Prof. Hans-Joachim Bungartz, who made this IDP possible.

I would also like to thank the Scientific Computing Chair for allowing me to present my work at the colloquium and the Mathematics Department for allowing me to take the Foundations of Data Analysis course, which pushed me out of my comfort zone and allowed me to gain a better understanding of many of the papers reviewed.

3 Introduction

One of the recent standard neural network architectures is the transformer, it is one of the most promising seq2seq models that have emerged. Its main feature is the attention mechanism, as opposed to previous models that were based on a recurrent neural network. It appears that as neural networks become larger, the more powerful and easier they become to analyze. In fact, in the infinite width limit, it is often possible to abstract away all the parameters of the neural network and make surprisingly strong closed statements about the behavior of the network. Currently, a major research direction in the field of theoretical deep learning focuses on the behavior of NNs given an infinite number of parameters in the system. It has been shown, and in particular for transformers, that in the limit of infinite width size, the distribution on the neural network for any set of data points becomes jointly Gaussian with a specific compositional covariance kernel: the NNGP kernel. Furthermore, these neural networks are equivalent to kernel-based regression using a family of new tangent neural kernels (NTKs). This proposes that a deeper insight into NNGP and NTK can drive new approaches to neural network analysis.

4 Definitions and notation

4.1 Heat kernel and Laplace-Beltrami operator

Let M be a Riemannian manifold embedded in $\mathbb{R}^n$.

Definition 1 (Laplace-Beltrami operator) *The intrinsic Laplace-Beltrami operator is defined as:*

$$\begin{aligned} \Delta : C^n(M) &\to C^n(M) \\ u &\mapsto \operatorname{div}(\nabla u). \end{aligned}$$

Manifold learning is defining the underlying manifold from a given amount of collection of measurements of a finite number of points on a manifold. Considerable consideration within the field of manifold learning has been devoted to providing a robust estimate of this Laplace-Beltrami operator on a manifold. When given data x_i sampled from a manifold M, such methods construct a graph whose individual weights are defined by a kernel function $k(x_i, x_j)$, thereafter approximating by the Laplacian of the graph, the Laplacian operator.

More precisely, if we aim to embed a smooth manifold $\mathcal{M} \subset \mathbb{R}^p$ so that at every $x \in \mathcal{M}$, the tangent space $T_x(\mathcal{M})$ is defined, in one-dimension $f : \mathcal{M} \to \mathbb{R}$. The tangent space inherits a local orthonormal coordinate system because every point $z \in \mathcal{M}$ has a unique closest point in $T_x(\mathcal{M})$ within some neighborhood.

While the local coordinate system is not unique and computing the gradient vector $\nabla f(x)$ depends on the choice of this system, the norm $\|\nabla f(x)\|$ is uniquely defined.

For any point $z \in \mathcal{M}$, one can show that

$$|f(z) - f(x)| \leq \|\nabla f(x)\| \|z - x\| + o(\|z - x\|)$$

Thus, to first order, $\|\nabla f\|$ measures how far apart f maps nearby points.

A reasonable objective is to minimize

$$\widetilde{\Phi}_{\text{lap}}(f) = \int_{\mathcal{M}} \|\nabla f\|^2 \quad \text{subject to} \quad \|f\| = 1,$$

since our goal is to find a map that best preserves locality on average.

Let $\Delta(f) = \sum_i \frac{\partial^2 f}{\partial z_i^2}$, where z_i are the tangent space coordinates, denote the Laplace-Beltrami operator on a manifold. It can be shown that

$$\widetilde{\Phi}_{\text{lap}}(f) = \int_{\mathcal{M}} \|\nabla f\|^2 = \int_{\mathcal{M}} \Delta(f) f$$

Thus, the function f minimizing $\widetilde{\Phi}_{\text{lap}}(f)$ must be an eigenfunction of the Laplace-Beltrami operator $\Delta(f)$, or equivalently a member of the null space of the following functional:

$$\mathcal{L}(f) = \int_{\mathcal{M}} (\Delta(f))^2$$

In particular, the normalized graph Laplacian $\mathcal{L}$ approximates the continuous Laplace-Beltrami operator Δ.[1]

The Laplacian-based approach to manifold learning is also justified by the fact that the Laplace-Beltrami operator encodes all geometric information of a Riemannian manifold. A demonstration of this fact follows from the product formula of the Laplacian

$$\Delta(fh) = f\Delta h + h\Delta f - 2\operatorname{grad} f \cdot \operatorname{grad} h, [2]$$

where the dot-product above is actually the Riemannian inner product $g_x : T_xM\times T_xM \to \mathbb{R}$

$$\begin{aligned} g_x(\operatorname{grad} f(x), \operatorname{grad} h(x)) &= (\operatorname{grad} f \cdot \operatorname{grad} h)(x) \\ &= \frac{1}{2}(f(x)\Delta h(x) + h(x)\Delta f(x) - \Delta(fh)(x)).[2] \end{aligned}$$

Since the geometry of a Riemannian manifold is entirely determined by the Riemannian metric, the above formulas show that the metric is entirely recoverable from the Laplacian.

The Laplacian operator captures the "geometry" in a broad sense. More concretely, the idea is that the scattering on a manifold is governed by the semigroup generated by the manifold's Laplace-Beltrami operator; the spectral analysis of the scattering operator thus provides information about the manifold that can be used to provide a lower-dimensional parameterization for the data. For example, by simulating a random walk or diffusion process on the manifold by taking small steps through the data set according to the probabilities estimated from the distances between data points. [3]

The above technique's reliance on capturing the manifold hinges on the observation that the optimal embedding for the manifold is achieved by the Laplace-Beltrami operator. And the central significance of that operator in the heat or diffusion equation indicates that the heat kernel provides a reliable means of selecting the weights for the manifold.

Definition 2 (Heat kernel [4]) *"A family $\{p_t\}_{t>0}$ of $\mu\otimes\mu$-measurable functions $p_t(x,y)$ on $M\times M$ is called a heat kernel if the following conditions are satisfied, for μ-almost all $x, y \in M$ and all $s, t > 0$:*

(i) Positivity: $p_t(x,y) \geq 0$.

(ii) Stochastic completeness:

$$\int_M p_t(x,y)d\mu(y) \equiv 1$$

(iii) Symmetry: $p_t(x,y) = p_t(y,x)$.

(iv) Semigroup property:

$$p_{s+t}(x,y) = \int_M p_s(x,z)p_t(z,y)d\mu(z)$$

(v) Approximation of identity: for any $u \in L^2$

$$\int_M p_t(x,y)u(y)d\mu(y) \xrightarrow{L^2} u(x) \qquad as\ t \to 0+.\text{"}$$

4.2 Neural Network gaussian process (NNGP) & Neural Tangent Kernel (NTK)

Definition 3 (Gaussian process [5]**)** *"We say a random function $f : X \to \mathbb{R}^m$ (with fixed dimensional output) is a Gaussian process if for any finite subset $\left\{x^1, \ldots, x^k\right\} \subseteq X$, the random vector $\left(f\left(x^1\right), \ldots, f\left(x^k\right)\right) \in \mathbb{R}^{m\times k}$ is distributed as a km-dimensional Gaussian. If f has variable dimensional output (e.g. f is an RNN), such as when $f(x) \in \mathbb{R}^{l(x)}$ for some length function $l : X \to \mathbb{N}^3$, then we say f is a Gaussian process if for any finite subset $\left\{x^1, \ldots, x^k\right\} \subseteq X$, the random vector $\left(f\left(x^1\right), \ldots, f\left(x^k\right)\right)$ is distributed as a $\left(\sum_i l\left(x^i\right)\right)$-dimensional Gaussian."*

The goal of a Gaussian process is to model an underlying distribution of $X = \left\{x^1, \ldots, x^k\right\}$ with $Y = \left\{y^1, \ldots, y^k\right\}$ as a multivariate normal distribution by treating each data entry point as a random variable and taking the corresponding multivariate normal distribution of dimension the number of entries. It is thus simply a matter of drawing samples from the joint probability distribution $P_{X,Y}$ that spans the space of potential values of the function for the one we want to predict.

NNGP A key result is that as the hidden layers of the neural network become infinitely large, the distribution of the features converges to a Gaussian process. Thus, the distribution on the neural network for any set of data points becomes jointly Gaussian with a particular compositional covariance kernel: the NNGP kernel. This result is important because this large-width bound is of practical interest, as neural networks of finite width generally perform strictly better as the width of the layers increases [6]. The correspondence between infinite-width neural networks and Gaussian processes for the specific case of a fully connected architecture is well studied, but it has also been demonstrated in [7] for standard neural networks containing standard layers such as convolution, pooling, connection hopping, attention, batch normalization, and/or layer normalization.

NN-GP correspondence for an MLP [5] Consider an MLP with widths $\left\{n^l\right\}_l$, weight matrices $\left\{W^l \in \mathbb{R}^{n^l \times n^{l-1}}\right\}_l$, and biases $\left\{b^l \in \mathbb{R}^{n^l}\right\}_l$, where l ranges among the layer numbers of the MLP. Its computation is given recursively as

$$h^1(x) = W^1 x + b^1 \quad \text{and} \quad h^l(x) = W^l \phi\left(h^{l-1}(x)\right) + b^l \text{ for } l \geq 2.$$

At initialization time, suppose $W_{\alpha\beta}^{l} \sim \mathcal{N}\left(0, \sigma_w^2/n^{l-1}\right)$ for each $\alpha \in \left[n^l\right], \beta \in \left[n^{l-1}\right]$, and $b_\alpha^l \sim \mathcal{N}\left(0, \sigma_b^2\right)$. Consider two inputs x, x'. Conditioned on $h^{l-1}(x)$ and $h^{l-1}\left(x'\right)$, iid for each α, $\left(h^l(x)_\alpha, h^l\left(x'\right)_\alpha\right)$ is distributed as

$$\mathcal{N}\left(0, \frac{\sigma_w^2}{n^{l-1}}\left(\begin{array}{cc} \left\|\phi\left(h^{l-1}(x)\right)\right\|^2 & \phi\left(h^{l-1}(x)\right)\cdot\phi\left(h^{l-1}\left(x'\right)\right) \\ \phi\left(h^{l-1}(x)\right)\cdot\phi\left(h^{l-1}\left(x'\right)\right) & \left\|\phi\left(h^{l-1}\left(x'\right)\right)\right\|^2 \end{array}\right) + \sigma_b^2.\right)$$

If $\left(h^{l-1}(x)_\alpha, h^{l-1}\left(x'\right)_\alpha\right)$ is distributed as $\mathcal{N}\left(0, \Sigma^{l-1}\right)$, iid for each α, then by a law of large number argument, the covariance matrix above converges to a deterministic limit

$$\Sigma^l \stackrel{\text{def}}{=} \sigma_w^2 \underset{\left(z, z'\right)\sim\mathcal{N}\left(0, \Sigma^{l-1}\right)}{\mathbb{E}}\left(\begin{array}{cc} \phi(z)^2 & \phi(z)\phi\left(z'\right) \\ \phi(z)\phi\left(z'\right) & \phi\left(z'\right)^2 \end{array}\right) + \sigma_b^2$$

as the width $n^{l-1} \to \infty$, making $\left(h^l(x)_\alpha, h^l\left(x'\right)_\alpha\right)$ Gaussian distributed as $\mathcal{N}\left(0, \Sigma^l\right)$. Iteratively applying this argument for each l yields the result for a deep MLP.

NTK [7][8][9] Given a parametrized function $f(x;\theta)$ with parameter θ and with scalar output, one can expand f in θ around a base point θ_0

$$f(x;\theta) - f\left(x;\theta_0\right) \approx \left\langle \nabla_\theta f\left(x;\theta_0\right), \theta - \theta_0\right\rangle$$

for any input x, where $\langle,\rangle$ denotes the inner product. $f(-;\theta) - f\left(-;\theta_0\right)$ is a linear model, where $\nabla_\theta f\left(-;\theta_0\right)$ acts as a input features, and $\theta - \theta_0$ acts as the weights. This approximation only works near θ_0. Considering f as a neural network, we can only train it shortly by gradient descent and with the restriction of a small learning rate to use this naive linearization. However, a breakthrough result is that if the widths of the network tend to infinity, f can fit any data perfectly and the naive expansion remains an accurate description of the learning dynamics. It could be that since the individual parameters move less and less with increasing width, it makes sense to do a Taylor expansion near the first parameter value. And the model itself is linear, so somehow a desperately complicated optimization path of an MLP has reduced to a kernel gradient descent with a fixed kernel: the NTK kernel, and in function space, this means that the predictions of the model become the same as a kernel machine. More precisely, consider the L-hidden-layer MLP $f(x;\theta)$ with width n^l in layer l. The finite-width NTK $\Theta(x, \bar{x}) \stackrel{\text{def}}{=} \left\langle \nabla_\theta f(x;\theta), \nabla_\theta f(\bar{x};\theta)\right\rangle$ converges in probability

$$\Theta \xrightarrow{\text{p}} \overset{\circ}{\Theta} \text{ as } n^1, \ldots, n^L \to \infty \text{ in that sequence,}$$

for some deterministic $\overset{\circ}{\Theta}$, over the randomness induced by randomly initializing the parameters like $\omega_{\alpha\beta}^{l}, b_{\alpha}^{l} \sim \mathcal{N}(0,1), \forall\alpha, \beta$. Thus, even with the random parameters θ, $\nabla_{\theta} f\left(x ; \theta_{0}\right), \nabla_{\theta} f\left(\bar{x} ; \theta_{0}\right)$ converges, in the infinite-width limit. Hence, the infinite-width NTK captures an implicit prior driven by gradient descent and choices of architecture and the initialization scheme. In particular, its spectrum provides information about the type of functions that can be learned quickly or that can be generalized well.

4.3 Transformer Network

A transformer [10] follows an Encoder-Decoder model. On one hand, the "encoding" consists of N encoders (N=6 in [10]) placed one after another: the encoder input is the previous encoder output, with the input of the first encoder being the vector embedding itself. The encoder is built up of two components: a feed-forward layer placed after a self-attention layer. The key feature is the self-attention layer which retains the interrelation of the words in the chosen representation of the sequence. Self-attention is the mechanism of attention used for a single sequence, to identify the interrelation of the individual words within the same sequence and to assign a relevant encoding to each of them. For instance, in the sentence: *The child is eating a sandwich because he is hungry*, it is obvious to a human being that *he* refers to *child* and not to *sandwich*. The process of self-attention will therefore aim at noticing the link between *child* and *he*.

On the other hand, the decoding unit is built around N decoders placed one after the other, with additional input coming from the last encoder. This means that the input of each decoder is formed by the previously encoded words combined with the output of the preceding decoder. Particularly, the latest decoder is directly interfaced with a "Linear Neural Network + Softmax" unit. The purpose of this unit is to identify which vocabulary words match the last encoder's output.

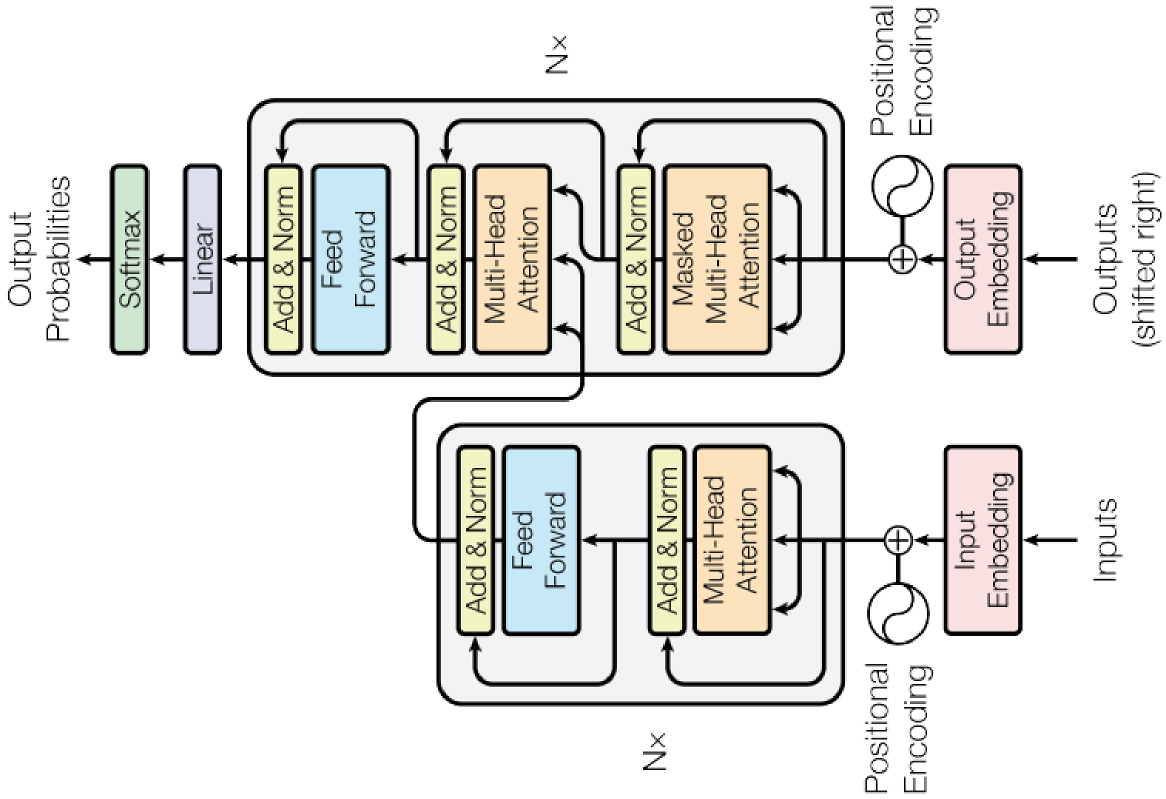

Figure 1: Transformer architecture [10]

5 Infinitely wide Transformer networks and their Laplace operators

5.1 Heat kernel family

One of the ideas that failed was to directly index the kernel corresponding to the neural network architecture by its depth so that one family of kernels could be considered for each architecture. Indeed, we sought to demonstrate all the properties of a heat kernel family as described in 4.1 for each family generated by classical neural network architecture. The selection of depth as the time indexation was inspired by the fact that the neural ordinary differential equation (ODE) approach in classical machine learning parameterizes the network according to its depth. However, this idea was not conclusive because, whereas some properties were verified (symmetry, semigroup properties), a fundamental non-constructive property, that is, the positive condition, typically does not hold for an NNGP or NTK kernel of a neural network.

5.2 Mapping to the Laplace-Beltrami operator

5.2.1 Spectra of the operator

A Hilbert-Schmidt kernel is a function $k : \Omega \times \Omega \to \mathbf{C}$ with a finite norm:

$$\int_\Omega \int_\Omega |k(x,y)|^2 dxdy < \infty;$$

this is true for all kernels corresponding to neural networks with standard architecture.

For each Hilbert-Schmidt kernel we can construct an associated Hilbert-Schmidt integral operator $K : L^2(\Omega; \mathbf{C}) \to L^2(\Omega; \mathbf{C})$ given by

$$(Ku)(x) = \int_\Omega k(x,y)u(y)dy$$

This operator is compact, and then the following assertions hold: [11]

(a) $\sigma(K) = \{0\} \dot{\cup} \{\lambda_j \mid j \in J\}$, where $J \in \{\varnothing, \mathbb{N}, \{1, \ldots, n\} \mid n \in \mathbb{N}\}$

(b) $\sigma(K) \backslash \{0\} = \sigma_p(K) \backslash \{0\}$. For all $\lambda \in \sigma(K) \backslash \{0\}$ the range of $\lambda I - K$ is closed and

$$\dim \mathrm{N}(\lambda I - K) = \operatorname{codim} \mathrm{R}(\lambda I - K) < \infty.$$

(c) For all $\varepsilon > 0$ the set $\sigma(K) \backslash B(0, \varepsilon)$ is finite. Hence, $\lambda_j \to 0$ as $j \to \infty$ if $J = \mathbb{N}$

As a result, the spectrum of the operator associated with the NNGP or NTK kernel that we will refer to in this work as neural operator K of a standard architecture neural network exhibits good behavior, and it is reasonable to expect a correspondence between this operator and the Laplace-Beltrami operator or between the NNGP/NTK kernel and the heat kernel.

5.2.2 Existence of orthogonal basis

The neural operator defined is bounded (because it is continuous), compact and self-adjoint. Therefore, by applying the Hilbert–Schmidt theorem we obtain the existence of a sequence of non-null real eigenvalues $\lambda_i, i = 1, \ldots, N$, where N equal to the rank of K, such that $|\lambda_i|$ is monotonically non-increasing and, if $N = +\infty$,

$$\lim_{i \to +\infty} \lambda_i = 0$$

Moreover, there exists an orthonormal set $\varphi_i, i = 1, \ldots, N$, and the functions φ_i form an orthonormal basis for the range of K and K can be written as

$$Ku = \sum_{i=1}^{N} \lambda_i \langle \varphi_i, u \rangle \varphi_i \text{ for all } u \in L^2(\Omega; \mathbf{C})$$

Regrettably, we need the neural operator range to be dense because it follows that the linear subspace spanned by the eigenvectors will be dense in $L^2(\Omega; \mathbf{C})$ and therefore the set is a complete orthogonal basis; but this is not a trivial assumption.

5.2.3 Transformation of bases of $L^2(\Omega; \mathbf{C})$

What we would like to achieve is to create equivalence classes between 2 bases of L^2 to link the orthogonal base defined by the neural operator (when this base exists) and the orthogonal base defined by the Laplace-Beltrami operator. Since this subject has hardly any literature already available, we already have to give up working with orthogonal bases in favor of working with frames that are more extensively studied due to the more restrictive nature of the framework of the orthogonal base. The characterization of the equivalence classes of $L^2(\Omega; \mathbf{C})$ frames that was chosen is as follows [12].

Let $\mathbf{I}$ be a countable index set. A family of vectors $\mathcal{F} = \{f_i\}_{i \in \mathbf{I}}$ in $L^2(\Omega; \mathbf{C})$ is called a (Hilbert) frame if there exist two real numbers $0 < A \leq B < \infty$ such that for any $x \in L^2(\Omega; \mathbf{C})$ we have:

$$A\|x\|^2 \leq \sum_{i \in \mathbf{I}} |< x, f_i >|^2 \leq B\|x\|^2$$

The analysis operator associated with $\mathcal{F}$ is defined by

$$T : L^2(\Omega; \mathbf{C}) \to l^2(\mathbf{I}), T(x) = \left(< x, f_i \rangle\right)_{i \in \mathbf{I}},$$

That operator is bounded and its norm is $\|T\| = \sqrt{B}$ and its range is closed.

Two frames $\mathcal{G} = \{g_i\}_{i \in \mathbf{I}}$ and $\mathcal{F} = \{f_i\}_{i \in \mathbf{I}}$ are said to be quadratically close if there exists a positive number $\lambda \geq 0$ such that:

$$\left\|\sum_{i\in\mathbf{I}} c_i\left(g_i-f_i\right)\right\| \leq \lambda\left\|\sum_{i\in\mathbf{I}} c_i f_i\right\|$$

for any $c=\left(c_i\right)_{i\in\mathbf{I}} \in l^2(\mathbf{I})$.

We note $c(\mathcal{G},\mathcal{F})$ the closeness bound of the frame $\mathcal{G}$ to the frame $\mathcal{F}$ and that bound is the infimum of the aforementioned λ 's.

If $\mathcal{G}$ is close to $\mathcal{F}$ with proximity bound less than 1, then $\mathcal{F}$ is also quadratically close to $\mathcal{G}$ but with different proximity bound, so this relation is not equivalent because it is not generally reflective. Consequently, it follows that

$$\left\|\sum_{i\in\mathbf{I}} c_i\left(g_i-f_i\right)\right\| \leq \frac{\lambda}{1-\lambda}\left\|\sum_{i\in\mathbf{I}} c_i g_i\right\|$$

To address the non-reflexivity of the close relation, we can assume two frames $\mathcal{F}=\left\{f_i\right\}_{i\in\mathrm{I}}$ and $\mathcal{G}=\left\{g_i\right\}_{i\in\mathrm{I}}$ to be close if $\mathcal{F}$ is close to $\mathcal{G}$ and $\mathcal{G}$ is close to $\mathcal{F}$. and in such a way it is convenient to set the pre-distance between $\mathcal{F}$ and $\mathcal{G}$, noted $d^0(\mathcal{F},\mathcal{G})$ as the greatest value between these two proximity bounds:

$$d^0(\mathcal{F},\mathcal{G})=\max(c(\mathcal{F},\mathcal{G}),c(\mathcal{G},\mathcal{F}))$$

A problem is that d^0 doesn't satisfy the triangle inequality so one can define the distance between $\mathcal{F}$ and $\mathcal{G}$ by:

$$d(\mathcal{F},\mathcal{G})=\log\left(d^0(\mathcal{F},\mathcal{G})+1\right)$$

As the relation of closeness is now an equivalence relation, we can partition the set of all frames on $L^2(\Omega;\mathbf{C})$, denoted $\mathcal{F}(L^2(\Omega;\mathbf{C}))$, into disjoint equivalent classes, indexed by a set of indices $\mathbf{A}$:

$$\mathcal{F}(L^2(\Omega;\mathbf{C}))=\bigcup_{\alpha\in A}\mathcal{E}_\alpha$$

with the following properties:

$$\mathcal{E}_\alpha\cap\mathcal{E}_\beta=\emptyset,\text{ for }\alpha\neq\beta$$

$\forall\mathcal{F},\mathcal{G}\in\mathcal{E}_\alpha, d(\mathcal{F},\mathcal{G})<\infty$ and $\forall\mathcal{F}\in\mathcal{E}_\alpha,\mathcal{G}\in\mathcal{E}_\beta$ with $\alpha\neq\beta, d(\mathcal{F},\mathcal{G})=\infty$

This distance divides the set of frames into closed subspaces of the space of coefficients $l^2(\mathbf{I})$ that are the equivalence classes.

The key result of [12] is that there exists a bounded and invertible operator on the Hilbert space that maps one frame set into the other if and only if two frames are at a finite distance (with the last distance defined) and this happens if and only if their analysis operators have the same closed range in $l^2(\mathbf{I})$.

The scope of the research for the presented work halted here as this was an unconstructive approach to transformation and the process of carrying it out appeared quite challenging.

5.3 Reproducing Kernel Hilbert Space

A proper way to express a kernel is by the associated reproducing kernel Hilbert space (RKHS) which reproduces a set of functions, and the resulting induced RKHS norm. That set is determined by the eigenfunctions and eigenvalues of the respective kernel under the uniform norm. The eigenvalue decay rate is of particular significance since it determines the smoothness properties of the functions. Nevertheless, knowing that the eigenvalues of two kernels decay at the same rate, is not sufficient in itself to state that the RKHS structure is identical. Even across different depths for the same kernel, the RKHS structure is not identical and produces different associated norms. As such, two kernels whose eigenvalues decay at the same rate can, and likely will, yield an outcome that is quite different when applied to the same regression problem.

In particular, one of the key results from which [13][14] derives is that of [15]. To examine the RKHS, a standard practice is to decompose, for some measure τ, the integral operator T given by $Tf(x) = \int k(x,y)f(y)d\tau(y)$ into its spectral components. The dot-product kernels of the form $k(x,x') = \kappa\left(x^\top x'\right)$ depend only on the angle between x and x' when inputs lie on the sphere $\mathbb{S}^{d-1}$. Rotation-invariant kernels can be diagonalized using spherical harmonics, leading to a link between the decay of eigenvalues and regularity. Indeed, using Mercer's theorem and for the particular case where τ is the uniform measure on $\mathbb{S}^{d-1}$, the RKHS $\mathcal{H}$ is given by

$$\mathcal{H} = \left\{ f = \sum_{k\geq 0,\mu_k\neq 0} \sum_{j=1}^{N(d,k)} a_{k,j} Y_{k,j}(\cdot) \quad \text{s.t.} \quad \|f\|_{\mathcal{H}}^2 := \sum_{k\geq 0,\mu_k\neq 0} \sum_{j=1}^{N(d,k)} \frac{a_{k,j}^2}{\mu_k} < \infty \right\}.$$

In particular, if

$$\mu_k = \frac{\omega_{d-2}}{\omega_{d-1}} \int_{-1}^{1} \kappa(t) P_k(t) \left(1-t^2\right)^{(d-3)/2} dt,$$

has a fast decay, then f's coefficient $a_{k,j}$ must also decay quickly with k for f to be in $\mathcal{H}$, and thus how we determine the regularity of f.

It was shown in [13][14], that the Laplace kernel and the respective neural tangent kernels of the fully connected networks (FC-NTK) and the residual fully connected networks with ReLU activation (ResNTK), for an input distributed uniformly over the hypersphere $\mathbb{S}^{d-1}$, have the same eigenfunctions (spherical harmonics) and their eigenvalues decrease polynomially with frequency k as k^{-d}. This implies that these three kernels have a close connection, which is not quite the goal we were aiming for since we intended to establish a connection with a heat kernel rather than with the Laplace kernel but is still a strong and notable result. Various researchers have indeed found that the Laplace kernel, defined as $\boldsymbol{k}^{\mathrm{Lap}}(\mathbf{x}, \mathbf{z}) = e^{-c\|\mathbf{x}-\mathbf{z}\|}$ for points $\mathbf{x}, \mathbf{z} \in \mathbb{S}^{d-1}$ and constant $c > 0$ performs similarly to neural networks when fitting data with gradient descent.

5.4 Transformer network: IMDb Movie Reviews

To determine the emotional tone of a text, sentiment analysis combines machine learning and natural language processing. In the IMDb dataset, movie reviews were labeled with positive and negative sentiment classifiers and the resulting dataset contained 50,000 reviews, which were split between positive (+1) and negative (-1) labels.

Our implementation was done in Neural Tangents [16][17][18][19][20] which is a high-level neural network API for specifying complex, hierarchical, finite, and infinite width neural networks using common building blocks like convolutions, pooling, residual connections, nonlinearities, but it is still a work in progress with some classical blocks that are yet to be implemented. For this task, we only need the encoder layer of the transformer network since our goal is to pass a sentence embedding vector to get a label output of -1 or +1, unlike other NLP tasks such as translation which requires a decoder to output back to a sentence. Although we always include at least one self-attention layer since it represents the very core of a transformer, we modify the other layers used and thus the resulting architecture, vis-à-vis the original paper, and within the limits of the layers available in the library. The embedding used is the classic GloVe, which is an unsupervised learning algorithm for deriving vector representations of words that use global word-word co-occurrence statistics.

All models were trained on 600 sentences and then evaluated on another set of 600 sentences for accuracy, with a maximum sentence length of 500. For more details on the implementation, please refer to the attached code and be sure to run only one model at a time. Understandably, the scores obtained from that small sample (roughly 1% of the total database) are not competitive at all with the current state of the art.

Model 1: Convolutional layer + Layernorm layer + Self-attention layer + Dropout layer + Normalization layer + Dropout layer + Normalization layer + Dense layer.

Model 2: Convolutional layer + Self-attention layer + Relu layer + Dropout layer + Normalization layer + Self-attention layer + Relu layer + Dropout layer + Normalization layer + Dense layer.

Model 3: Convolutional layer + Relu layer + Self-attention layer + Relu layer + Normalization layer + Dense layer.

Model 4: Convolutional layer + Relu layer + Self-attention layer + Relu layer + Self-attention layer + Relu layer + Self-attention layer + Relu layer + Normalization layer + Dense layer.

Table 1: Accuracy and loss for various attention encoder models represented by their NNGP kernels

	Model 1	Model 2	Model 3	Model 4
Network Accuracy	0.610	0.7	0.743	0.753
Network Loss	0.121	0.098	0.092	0.0935

Table 2: Accuracy and loss for various attention encoder models represented by their NTK kernels

	Model 1	Model 2	Model 3	Model 4
Network Accuracy	0.610	0.716	0.742	0.732
Network Loss	0.121	0.094	0.09	0.09

Besides the standard purpose of examining the accuracy of the model on a dataset, we also investigate the trainability. Indeed, one of the long-lived goals of deep learning theory is to describe the conditions under which certain neural network architectures will be trainable, and this property analysis simplifies considerably within the limit of very wide and very deep networks. It was found [21] that deep neural networks can exhibit a phase transition as a function of the variance of their weights (σ_w^2) and biases (σ_b^2). Consider two normalized inputs to a neural network, x_1 and x_2 such that $\|x_1\| = \|x_2\| = q^0$. The cosine-angle between the inputs is $c^0 = \cos\theta_{12} = \frac{x_1 \cdot x_2}{q^0}$.As the signal passes through layers of the neural network, we can keep track of the norm q^l and the cosine angle c^l. In the wide-network limit there are deterministic functions, called the $\mathcal{Q}$-map and the $\mathcal{C}$-map, such that $q^{l+1} = \mathcal{Q}(q^l)$ and $c^{l+1} = \mathcal{C}(q^l, c^l)$. In classical networks and depending on the activation functions, both the $\mathcal{Q}$-map and $\mathcal{C}$-map have unique stable-fixed-points, q^* and c^*, such that $q^* = \mathcal{Q}(q^*)$ and $c^* = \mathcal{C}(q^*, c^*)$. For simplification, a typical choice is to normalize the inputs so that $q^0 = q^*$ and therefore restrict our study to the $\mathcal{C}$-map. The $\mathcal{C}$-map always has a fixed point at $c^* = 1$ since two identical inputs will remain identical as they pass through the network and the phase boundary is defined as the point where $c^* = 1$ is marginally stable.

In practice, for an implemented attention layer, we see on the phase diagram that σ_b^2 plays no role in the trainability when in equilibrium, whereas we have a very strict

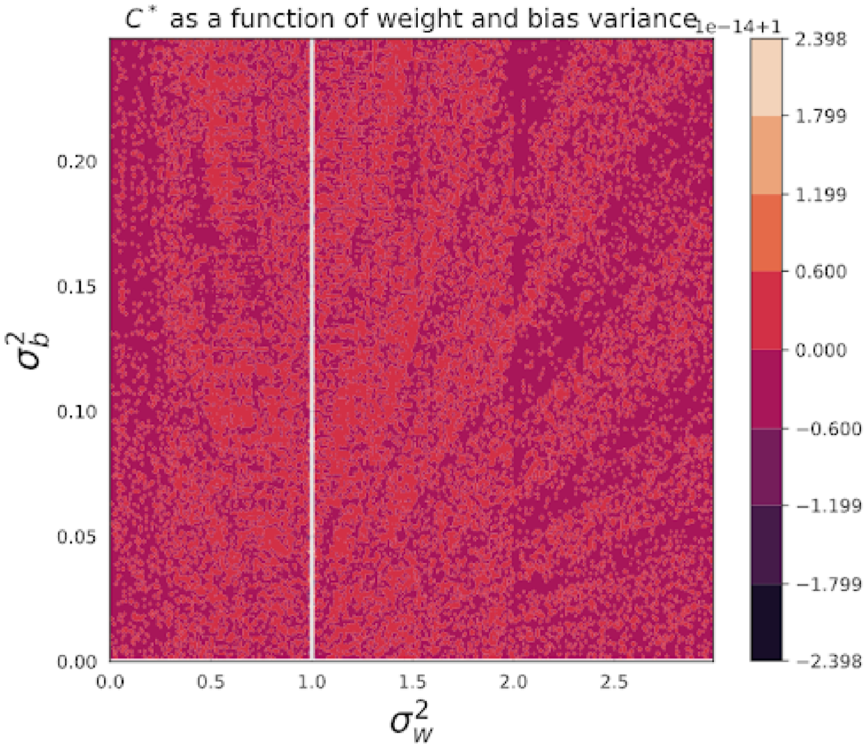

Figure 2: Phase diagram for an attention layer

requirement of having $\sigma_w^2 = 1$. And it's rather interesting to note that indeed, in the implementation in the google tangent library, they set the specification of σ_b^2 as optional for this layer whereas the default value of σ_w^2 is 1.

6 Conclusions

In section 5.1, we attempted to make a direct connection between neural network kernels and the family of heat kernels, the latter being closely related to the Laplace-Beltrami operator, but could not succeed because we lacked fundamental properties of heat kernels. In Section 5.2, we found that the eigenvalues of the neural operator defined as an integral kernel operator associated with a neural network kernel, behaved well; and subsequently, that under the dense range assumption, we could derive an orthogonal family defined by this neural operator. Afterward, we tried to find a transformation that could map this orthogonal family to a basis generated by a heat kernel, this task is still pending as the mapping is not yet straightforward. In section 5.3 we studied one of the current states of the art approach that proves similar smoothness properties of functions in the function sets of the reproducing kernel Hilbert space of the neuronal tangent kernel for two standard architectures and the RKHS of the Laplace kernel. In section 5.4, we deploy the NNGP and NTK Transformer kernels on an IMDb movie dataset for sentiment analysis and perform a quick phase diagram analysis for the attention layer

Going back to section 5.3, we may be able to generalize such a SOTA approach in the following two directions.

First, studying NNGP instead of NTK should be straightforward as there is a key result of NTK that also holds for NNGP. Indeed in [15] the deep NTK decay result: "For the neural tangent kernel κ^L_{NTK} of an L-layer ReLU network with $L \geq 3$, we have $\mu_k \sim C(d, L)k^{-d}$, where $C(d, L)$ is different depending on the parity of k and grows quadratically with L" has the NNGP counterpart "For the random neuron kernel κ^L_{RF} of an L-layer ReLU network with $L \geq 3$, we have $\mu_k \sim C(d, L)k^{-d-2}$, where $C(d, L)$ is different depending on the parity of k and grows linearly with L."

Second, by extending the result to other standard architectures, including the Transformer that interests us in this work. While it is almost trivial to find the eigenfunction of a large choice of standard architectures, one only has to show that the corresponding kernel is homogeneous of degree 1 and zonal to obtain that the eigenfunctions are spherical harmonics; sadly it is not straightforward to compute the decay rate of the eigenvalues of the kernels, and we can not underestimate the effort required to derive it.

In practice, it is clear that the manifold is essential, but the density of the measurements also matters. For example, if there are regions with zero density, they will be unrecoverable. Even with positive density everywhere, much more complex problems can arise and we often need the manifold to be sampled very densely. This is where the diffusion map algorithm proves excellent, as it removes the influence of sampling density. To grasp the general implications of the success of this work, we can see that a more distant goal based on this report would be to mimic the diffusion map algorithm for infinitely large neural networks and therefore to be able to extract an operator that matches a specific architecture independently of the data we present to it on the manifold (but with a required positive density); in other words, to factor out the density of the sampling data. It was therefore a natural choice that the weapon of choice for our approach was the Laplace-Beltrami operator since an important feature of the theory of diffusion maps is that it recovers the Laplace-Beltrami operator when the data set approaches a Riemann submodel.

References

[1] Erik Sudderth. *Nonlinear Manifold Learning Part II.* Lecture notes. MIT, 2002.

[2] Tyrus Berry and Dimitrios Giannakis. "Spectral exterior calculus". In: *Communications on Pure and Applied Mathematics* 73.4 (2020), pp. 689–770.

[3] Shan Shan and Ingrid Daubechies. "Diffusion Maps: Using the Semigroup Property for Parameter Tuning". In: *arXiv preprint arXiv:2203.02867* (2022).

[4] Alexander Grigor'yan and Jiaxin Hu. "Heat kernels and Green functions on metric measure spaces". In: *Canadian Journal of Mathematics* 66.3 (2014), pp. 641–699.

[5] Greg Yang. "Wide feedforward or recurrent neural networks of any architecture are gaussian processes". In: *Advances in Neural Information Processing Systems* 32 (2019).

[6] Roman Novak et al. "Bayesian deep convolutional networks with many channels are gaussian processes". In: *arXiv preprint arXiv:1810.05148* (2018).

[7] Greg Yang. "Tensor programs ii: Neural tangent kernel for any architecture". In: *arXiv preprint arXiv:2006.14548* (2020).

[8] Arthur Jacot et al. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems* 31 (2018).

[9] Greg Yang and Hadi Salman. "A fine-grained spectral perspective on neural networks". In: *arXiv preprint arXiv:1907.10599* (2019).

[10] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[11] Roland Schnaubelt. *Spectral Theory.* Lecture notes. 2020.

[12] Radu Balan. "Equivalence relations and distances between Hilbert frames". In: *Proceedings of the American Mathematical Society* 127.8 (1999), pp. 2353–2366.

[13] Yuval Belfer et al. "Spectral analysis of the neural tangent kernel for deep residual networks". In: *arXiv preprint arXiv:2104.03093* (2021).

[14] Amnon Geifman et al. "On the similarity between the laplace and neural tangent kernels". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1451–1461.

[15] Alberto Bietti and Francis Bach. "Deep equals shallow for relu networks in kernel regimes". In: *arXiv preprint arXiv:2009.14397* (2020).

[16] Roman Novak et al. “Neural Tangents: Fast and Easy Infinite Neural Networks in Python”. In: *International Conference on Learning Representations*. 2020. URL: `https://github.com/google/neural-tangents`.

[17] Roman Novak et al. “Fast Finite Width Neural Tangent Kernel”. In: *International Conference on Machine Learning*. 2022. URL: `https://github.com/google/neural-tangents`.

[18] Jiri Hron et al. “Infinite attention: NNGP and NTK for deep attention networks”. In: *International Conference on Machine Learning*. 2020. URL: `https://github.com/google/neural-tangents`.

[19] Jascha Sohl-Dickstein et al. *On the infinite width limit of neural networks with a standard parameterization*. 2020. URL: `https://github.com/google/neural-tangents`.

[20] Insu Han et al. *Fast Neural Kernel Embeddings for General Activations*. 2022. URL: `https://github.com/google/neural-tangents`.

[21] Lechao Xiao et al. “Disentangling trainability and generalization in deep neural networks”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 10462–10472.