The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)

Avoiding Kernel Fixed Points: Computing with ELU and GELU Infinite Networks

Russell Tsuchida,[12] **Tim Pearce,**[3] **Chris van der Heide,**[2] **Fred Roosta,**[24] **Marcus Gallagher**[2]

[1]CSIRO, [2]The University of Queensland, [3]University of Cambridge, [4]International Computer Science Institute

Abstract

Analysing and computing with Gaussian processes arising from infinitely wide neural networks has recently seen a resurgence in popularity. Despite this, many explicit covariance functions of networks with activation functions used in modern networks remain unknown. Furthermore, while the kernels of deep networks can be computed iteratively, theoretical understanding of deep kernels is lacking, particularly with respect to fixed-point dynamics. Firstly, we derive the covariance functions of multi-layer perceptrons (MLPs) with exponential linear units (ELU) and Gaussian error linear units (GELU) and evaluate the performance of the limiting Gaussian processes on some benchmarks. Secondly, and more generally, we analyse the fixed-point dynamics of iterated kernels corresponding to a broad range of activation functions. We find that unlike some previously studied neural network kernels, these new kernels exhibit non-trivial fixed-point dynamics which are mirrored in finite-width neural networks. The fixed point behaviour present in some networks explains a mechanism for implicit regularisation in overparameterised deep models. Our results relate to both the static iid parameter conjugate kernel and the dynamic neural tangent kernel constructions[1].

1 Background — Infinitely Wide Neural Networks as Gaussian Processes

Infinitely wide neural networks (NNs) and Gaussian processes (GPs) share an interesting connection (Neal 1995; Jacot, Gabriel, and Hongler 2018) which has only partially been explored. We begin by reviewing this connection. Readers familiar with this connection may skip to § 2. Consider a one-hidden layer network with independent parameters. Suppose each ith row of weights $\mathbf{W}_i$ together with the corresponding bias B_i in the hidden layer has distribution $(\mathbf{W}_i^\top, B_i)^\top = \widetilde{\mathbf{W}}_i \sim \mathcal{N}(\mu, \Sigma)$, with $\Sigma \succ 0$ being a diagonal matrix having a unique "square root" $\Sigma^{(1/2)}$. Further, suppose the output layer parameter vector $\mathbf{V} = \frac{1}{\sqrt{n}}\mathbf{U}$ satisfies $\mathbf{U} \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 I)$, where n is the number of neurons in the hidden layer and the output bias satisfies $V_b \sim \mathcal{N}(0, \sigma_b^2)$. The output evaluated at input $\mathbf{x}_1$ is $f(\mathbf{x}_1) = \frac{1}{\sqrt{n}} \sum_{i=1} U_i \psi(\widetilde{\mathbf{W}}_i^\top \widetilde{\mathbf{x}}_1) + V_b$, where ψ is an activation function and $\widetilde{\mathbf{x}}_1 = (\mathbf{x}_1^\top, 1)^\top$. The covariance between any two outputs is

$$
\begin{aligned}
&k^{(1)}(\mathbf{x}_1, \mathbf{x}_2) \\
&= \mathbb{E}\Big[\sum_{i=1}^n V_i \psi(\widetilde{\mathbf{W}}_i^\top \widetilde{\mathbf{x}}_1) \sum_{j=1}^n V_j \psi(\widetilde{\mathbf{W}}_j^\top \widetilde{\mathbf{x}}_2)\Big] + \sigma_b^2 \\
&= \sigma_w^2 \mathbb{E}\Big[\psi(\widetilde{\mathbf{W}}_1^\top \widetilde{\mathbf{x}}_1)\psi(\widetilde{\mathbf{W}}_1^\top \widetilde{\mathbf{x}}_2)\Big] + \sigma_b^2.
\end{aligned}
$$

The expectation over $d+1$ random variables reduces to an expectation over 2 random variables, $\widetilde{\mathbf{W}}_1^\top \widetilde{\mathbf{x}}_1$ and $\widetilde{\mathbf{W}}_1^\top \widetilde{\mathbf{x}}_2$. The joint distribution of these two random variables is a bivariate Gaussian. The mean of each component is zero, and the variance is $\|\Sigma^{(1/2)}\widetilde{\mathbf{x}}_i\|^2$, where $\|\cdot\|$ denotes the Euclidean norm. The covariance is $\|\Sigma^{(1/2)}\widetilde{\mathbf{x}}_1\|\|\Sigma^{(1/2)}\widetilde{\mathbf{x}}_2\|\cos\theta$, where θ is the angle between $\Sigma^{(1/2)}\widetilde{\mathbf{x}}_1$ and $\Sigma^{(1/2)}\widetilde{\mathbf{x}}_2$. Therefore, the expectation in terms of $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, S)$ is

$$
k^{(1)}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_w^2 \mathbb{E}\Big[\psi\big(s_1 Z_1 + \widetilde{\mu}_1\big)\psi\big(s_2 Z_2 + \widetilde{\mu}_2\big)\Big] + \sigma_b^2, \tag{1}
$$

where S has diagonals 1 and off-diagonals $\cos\theta$, $s_i = \|\Sigma^{(1/2)}\widetilde{\mathbf{x}}_i\|$ and $\widetilde{\mu}_i = \boldsymbol{\mu}^\top \widetilde{\mathbf{x}}_i$.

Definition 1. *We call* (1) *the kernel. We call* $\cos\theta^{(1)} = \frac{k^{(1)}(\mathbf{x}_1,\mathbf{x}_2)}{\sqrt{k^{(1)}(\mathbf{x}_1,\mathbf{x}_1)k^{(1)}(\mathbf{x}_2,\mathbf{x}_2)}}$ *the normalised kernel.*

The above NN converges to a GP as $n \to \infty$ under mild conditions on the input and activation function ψ (Neal 1995). Since $f(\mathbf{x}_1)$ is a sum of independent random variables scaling as $n^{-1/2}$, it converges to a Gaussian random variable with zero mean as $n \to \infty$. More generally, any fixed collection of N evaluations of f, $\{f(\mathbf{x}_i)\}_{i=1}^N$ converges to an N-dimensional $\mathbf{0}$-mean Gaussian as $n \to \infty$.

Analytical and closed-form covariance functions (1) are available for specific choices of ψ (Le Roux and Bengio 2007; Tsuchida, Roosta, and Gallagher 2018, 2019a; Pearce et al. 2019; Tsuchida, Roosta, and Gallagher 2019b), although some of these require $\boldsymbol{\mu} = 0$. Most notably, the kernel is known for historically relevant activation functions $\psi(z) = \text{erf}(z)$, RBF networks (Williams 1997) and

[1] Software at github.com/RussellTsuchida/ELU_GELU_kernels

for the more modern ReLU activation, $\psi(z) = \max(0, z)$ (Cho and Saul 2009). More recently Meronen, Irwanto, and Solin (2020) solved the *inverse* problem, finding ψ that recovers the Matérn class of covariance functions. Once the form of (1) is known, the kernel of deep networks can be evaluated in the case where $\Sigma = \operatorname{diag}(\sigma_w^2, ..., \sigma_w^2, \sigma_b^2)$ and $\boldsymbol{\mu} = \mathbf{0}$ (Matthews et al. 2018; Lee et al. 2018; Yang 2019b,a). The case where $\boldsymbol{\mu} \neq 0$ can also be handled (Tsuchida, Roosta, and Gallagher 2019b), but we focus on $\boldsymbol{\mu} = 0$ in this work. The kernel in layer $l+1$ can be found iteratively as a function of the kernel in layer l,

$$k^{(l+1)}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_w^2 \mathbb{E}\Big[\psi\big(s_1^{(l)} Z_1^{(l)}\big)\psi\big(s_2^{(l)} Z_2^{(l)}\big)\Big] + \sigma_b^2,$$

$$\begin{bmatrix} Z_1^{(l)} \\ Z_2^{(l)} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} 1 & \cos\theta^{(l)} \\ \cos\theta^{(l)} & 1 \end{bmatrix}\right), \qquad (2)$$

where $\cos\theta^{(l)}$ is the normalised kernel in layer l, and $s_i^{(l)} = \sqrt{k^{(l)}(\mathbf{x}_i, \mathbf{x}_i)}$.

Definition 2. *We call $k^{(l)}$ in* (2) *the kernel in layer l. We call $\cos\theta^{(l)} = \frac{k^{(l)}(\mathbf{x}_1, \mathbf{x}_2)}{\sqrt{k^{(l)}(\mathbf{x}_1, \mathbf{x}_1) k^{(l)}(\mathbf{x}_2, \mathbf{x}_2)}}$ the normalised kernel in layer l.*

A generalisation of iid weight priors to partially exchangeable weight priors is also available (Tsuchida, Roosta, and Gallagher 2019b), resulting in a GP with an additional layer of inference over the hyperparameters $\boldsymbol{\mu}$ and Σ. Convergence to GPs also occurs for other NN architectures such as convolutional architectures (Garriga-Alonso, Rasmussen, and Aitchison 2018; Novak et al. 2019) and general compositions of recurrent, graph convolution, pooling, skip connection, attention and normalisation layers (Yang 2019b,a)[2]. When an MLP is trained under a continuous-time analogue of gradient descent, the limiting output is still a GP (Jacot, Gabriel, and Hongler 2018). The dynamics and associated covariance of the GP depends on the neural tangent kernel (NTK) $T^{(l)}$, which in addition to the iterations (2), is given by

$$\begin{aligned} T^{(1)}(\mathbf{x}_1, \mathbf{x}_2) &= k^{(1)}(\mathbf{x}_1, \mathbf{x}_2) \\ \dot{k}^{(l+1)}(\mathbf{x}_1, \mathbf{x}_2) &= \sigma_w^2 \mathbb{E}\Big[\psi'(s_1 Z_1^{(l)})\psi'(s_2 Z_2^{(l)})\Big], \\ T^{(l+1)}(\mathbf{x}_1, \mathbf{x}_2) &= T^{(l)}(\mathbf{x}_1, \mathbf{x}_2)\dot{k}^{(l+1)}(\mathbf{x}_1, \mathbf{x}_2) \\ &\quad + k^{(l+1)}(\mathbf{x}_1, \mathbf{x}_2). \qquad (3) \end{aligned}$$

2 Contributions and Motivation

This paper contains two main contributions. We:

1. Derive kernels for GELU and ELU activations (defined below) and verify our results numerically. We implement GPs with different NN kernels on some benchmarks.
2. Study the fixed point dynamics of the kernel when ψ is bounded by the absolute value of a polynomial. We find sufficient conditions for the existence of a unique kernel fixed point. We show theoretically and empirically that unlike the kernel corresponding to ReLU ψ, the new kernels are able to avoid unique fixed points. These conditions apply to both the iid prior (Neal 1995) and dynamic NTK (Jacot, Gabriel, and Hongler 2018) cases. This fixed point behaviour can be used to explain a simplicity bias in deep NN. More surprisingly, we find theoretically that the NTK dynamic which approximates gradient descent preserves this simplicity bias.

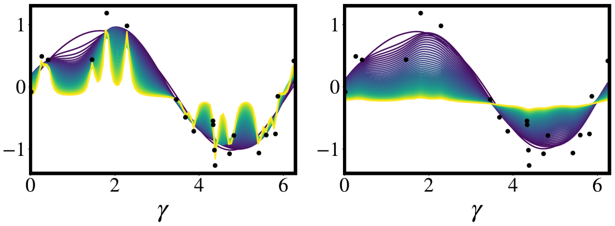


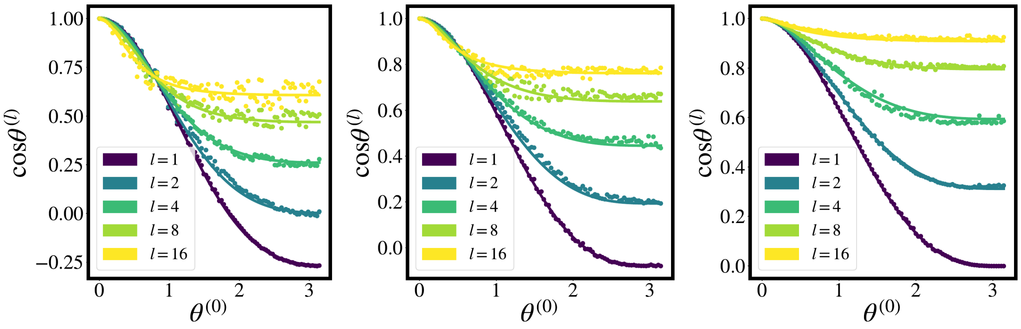
Figure 1: Illustration of simplicity bias due to kernel fixed points. Training data $\mathbf{x} \in \mathbb{R}^2$ is uniformly sampled on the unit disc at heading γ. Curves show the posterior mean of a GP regression model on $y = \sin(\gamma) + \epsilon$ with known additive noise variance. σ_w is chosen according to Figure 3. Colours move from purple to yellow as depth increases from 1 to 64. (Left) GELU without unique kernel fixed point leading to overfitting (Right) ReLU with unique kernel fixed point leading to underfitting. More examples in Appendix M.

2.1 Motivation for Studying Fixed Points

Viewing NNs through the arguably idealised lens of GPs has some surprisingly non-intuitive practical implications. One important open problem is in explaining the empirical observation that some overparameterised NNs do not overfit, even when trained without explicit regularisation (Zhang et al. 2017). Tsuchida, Roosta, and Gallagher (2019b) show empirically that samples from the limiting prior of deep MLPs with zero-mean parameters and LReLU activations are approximately constant on the unit hypersphere. Valle-Pérez, Camargo, and Louis (2019) argue that deep NNs with ReLU activations exhibit a "simplicity bias", in that randomly initialised NNs implementing Boolean functions are likely to be simple. Yang and Salman (2019) explain this simplicity bias through a spectral decomposition of the limiting kenel, showing that most of its mass is concentrated around the constant eigenfunction, even when accounting for training using gradient descent under the NTK (Jacot, Gabriel, and Hongler 2018). Yang and Salman (2019) are clear to separate the case of ReLU activations, which do result in kernels having peaked spectral mass and exhibiting a simplicity bias, from ERF activations which do not.

Our motivation for studying fixed points is in a similar spirit to the work above. For LReLU networks, we observe a so called kernel fixed point. An infinitely deep LReLU network is degenerate, and therefore over-regularised, in that all functions in the prior are constant over inputs on any hypersphere. Therefore, increasingly deep kernels rep-

[2] As detailed in these works, knowledge of (1) is often sufficient to describe these kernel, so our new kernels apply to more complicated networks.



$$k(\mathbf{x}_1, \mathbf{x}_2) =$$

$$\sigma_b^2 + \sigma_w^2 \left(\frac{s_1 s_2}{4} \cos\theta + \frac{s_1^2 s_2^2}{2\pi} \left[\frac{\frac{1}{2}(\cos(2\theta) + 3) + s_1^2 + s_2^2 + s_1^2 s_2^2 \sin^2\theta}{(1+s_1^2)(1+s_2^2)\sqrt{1 + s_1^2 + s_2^2 + s_1^2 s_2^2 \sin^2\theta}} + \frac{\cos\theta}{s_1 s_2} \tan^{-1}\left(\frac{\cos\theta s_1 s_2}{\sqrt{1 + s_1^2 + s_2^2 + s_1^2 s_2^2 \sin^2\theta}} \right) \right] \right)$$

Figure 2: The GELU kernel. Plots show the normalised kernels in layer l as a function of the angle $\theta^{(0)}$ between the inputs for MLPs of increasing depth when $\Sigma = \text{diag}(\sigma_w^2, ..., \sigma_w^2, 0)$ and $\|\mathbf{x}_i\|$ is constant for all i. Values of σ_w are chosen to preserve the expected square norm $\|\mathbf{x}\|^2$ (see § 4.1). Solid curve shows infinitely wide limit, and dots show samples from a network with 2 inputs and 3000 neurons in each layer. Each dot corresponds to an $\mathbf{x}_1$ and $\mathbf{x}_2$ generated through a random rotation of $(1, 0)^\top$ and $(\cos\theta^{(0)}, \sin\theta^{(0)})^\top$. The random rotation is found through a QR decomposition of a matrix containing entries sampled independently from $\mathcal{U}[0, 1]$. (Left) $\|\mathbf{x}\| = 0.5$, $\sigma_w = 1.59$ (Middle) $\|\mathbf{x}\| = 1$, $\sigma_w = 1.47$. (Right) $\|\mathbf{x}\| = 5$, $\sigma_w = 1.42$.

resent a strict and potentially undesirable prior. On the other hand, kernels corresponding to GELU and ELU activation functions do not exhibit unique fixed points, and are therefore less biased towards simple functions. Just as traditional regularisation frameworks allow practitioners to control the bias-variance trade-off, in our framework, the activation function represents a similar choice. A deep ReLU network contains a higher degree of implicit regularisation than a deep GELU network. An illustration of this effect is shown in Figure 1. Even more surprisingly, our analysis extends to the NTK.

2.2 Recently Introduced Activation Functions

The increased volume of gradient-based deep learning research has seen the introduction of new popular activation functions. Notably these include the exponential linear unit (ELU) (Clevert, Unterthiner, and Hochreiter 2016), the Gaussian error linear unit (GELU) (Hendrycks and Gimpel 2016) and the Swish (Ramachandran, Zoph, and Le 2017; Elfwing, Uchibe, and Doya 2018). The GELU and ELU are

$$\psi(z) = z\Phi(z) \text{ and } \psi(z) = \Theta(z)z + \Theta(-z)(e^z - 1),$$

respectively, where Φ denotes the CDF of the standard Gaussian and Θ denotes the Heaviside step function.

Many state-of-the-art models use GELU (Radford et al. 2018; Devlin et al. 2019) or swish activations (Chua et al. 2018). However, even when critically evaluating empirical evidence, it is difficult to determine whether certain activation functions are a better choice for a given problem, let alone separate the activation function expressivity from the ability of optimisers to find good solutions. Analysing activation functions through the lens of GPs allows one to visualise the function space in isolation of the ability of the optimiser to find good solutions, and reveals interesting implicit regularisation structure in the infinitely wide setting.

2.3 Model Selection

The choice of activation function in an NN can be framed as a model selection problem, and as such shares similarities with choosing other model hyperparameters. Take the case of choosing the L2 ridge-regularisation parameter as an example. One could apply n-fold cross-validation, with an implicit understanding that this parameter penalises model complexity as measured through the norm in an RKHS. Alternatively, a Bayesian might interpret the L2 weighting as the precision of a Gaussian prior, and optimise the marginal likelihood with respect to this weighting. A more pure Bayesian might put a prior over the regularisation (or precision) parameter and marginalise over these models. Common to all these approaches is (a) an intuition based on theory of what the parameter does and (b) a practical methodology for dealing with the parameter. In this paper we provide (a), and leave it to the practitioner to decide on (b) based on their philosophy and/or apparatus. Our work shows that GELU and ELU activations can avoid a regularisation mechanism that grows with depth that is *always* implicit in ReLU activations.

We stress that the new kernels and the fixed point mechanisms are neither "good" nor "bad" in isolation. The new models should be evaluated according to model selection criteria in their given application, and the fixed point mechanisms describe a regularisation implicit in some kernels.

3 New Kernels

Proposition 3. *When ψ is the GELU and $\boldsymbol{\mu} = 0$, the kernel* (1) *is given by the equation in Figure 2.*

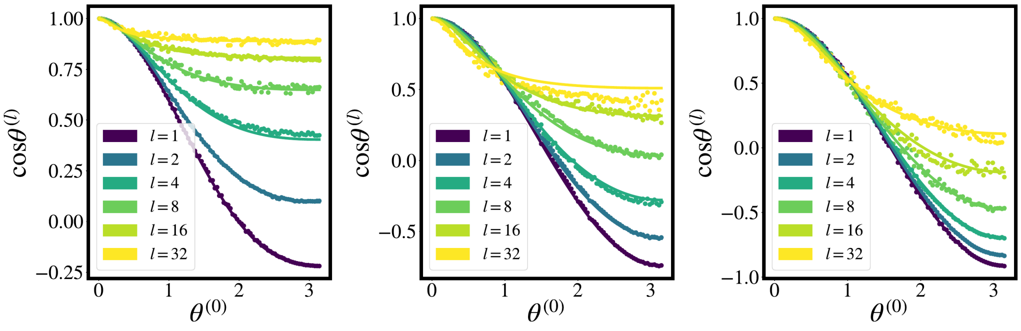


Figure 3: As in Figure 2, but with ELU ψ. (Left-Right) $(\|\mathbf{x}\|, \sigma_w) = (5, 1.40)$, $(1, 1.26)$, $(0.5, 1.17)$.

The derivation is in the same spirit as Williams (1997); we introduce dummy parameters β_1 and β_2 in the argument of Φ, differentiate with respect to β_1 and β_2 to obtain a PDE, then solve the PDE and evaluate the solution at $\beta_1 = \beta_2 = 1$. Complete working is given in Appendix A. It is plausible that our method of derivation extends to the case $\boldsymbol{\mu} \neq 0$, although the calculations and resulting expression become more complicated. Even with $\boldsymbol{\mu} = 0$, this kernel has some interesting properties that we discuss in § 4. Interestingly, unlike the ELU kernel with $\boldsymbol{\mu} = 0$, the GELU kernel does not contain any hard-to-compute special functions, only some (inverse) trigonometric functions.

Our expression for the ELU kernel is lengthy and we do not assemble it in the main text, but provide a visualisation in the form of Figure 3.

Proposition 4. *When ψ is the ELU, the kernel* (1) *has an analytical expression implemented in software*[1] *in terms of the univariate and bivariate normal CDFs.*

Complete working is given in Appendix B. Unfortunately, the ELU kernel involves exponentiating arguments involving s_1 and s_2. This can lead to numerical instability in GP regression when many data points are involved. Despite this, having an analytical expression still allows us to gain insights into finite width networks, as we shall see in § 4.

The scaled exponential linear unit (SELU) (Klambauer et al. 2017) is a slightly modified version of the ELU which introduces two scaling parameters: one applied over the entire domain and one over the negative domain. Our analysis for the ELU also handles the SELU (see Appendix B). Klambauer et al. (2017) motivates the SELU by showing that it is able avoid the exploding and vanishing gradient problem by carefully selecting the value of the scaling parameters. An entirely distinct problem is to analyse the *correlations* between signals in the network. Fixed points in signal norms are desirable to maintain the scale of signals as they propagate through the network. Fixed points in signal correlations can be undesirable as they force unrelated inputs to have similar feature representations in deep layers of the network. While Klambauer et al. (2017) is concerned with obtaining fixed points of signal norms (i.e. our § 4.1), it does not relate to fixed points of signal correlations (i.e. our § 4.2).

4 Fixed Point Analysis

In this section, we first analyse the conditions under which the expected squared norm of the signals in each layer are preserved as the signal propagates through the network for a different choices of the activation function (§ 4.1). In such a situation, the expected squared norm remains at a fixed point as it passes through the network. Then, more generally, we analyse conditions under which the expected squared norm of any two signals *and* the cosine angle between them approaches a constant as depth increases (i.e., when the kernel has a fixed point). We are especially interested in the case where the cosine angle between the signals converges to a unique fixed point (§ 4.2). Finally, we relate the existence of a unique fixed point to a degenerate, underfitting property of very deep infinitely wide MLPs (§ 4.3).

For § 4, 5 and 6, we suppose all the weights have the same variance, and so do all the biases; the first d diagonals of the diagonal matrix Σ are σ_w^2, and the last diagonal is σ_b^2.

4.1 Warm-Up — Norm Preservation

A useful application of the kernel in finite-width iid-initialised networks is to track the expected squared norm of the signals in each layer as the depth of the network increases. This is used in initialisation to avoid exploding or vanishing signals when using gradient optimisers.

The expected norm squared of the signal in the first hidden layer is $(k^{(1)}(\mathbf{x}_1, \mathbf{x}_1) - \sigma_b^2)/\sigma_w^2$. For the squared norm of the signal in the hidden layer to be the same as the squared norm of the input, we set $\|\widetilde{\mathbf{x}}_1\|^2 = (k^{(1)}(\mathbf{x}_1, \mathbf{x}_1) - \sigma_b^2)/\sigma_w^2$. We may then solve this condition to find the hyperparameter values that preserve input norms. For example, using the kernel corresponding to ReLU (Cho and Saul 2009), one obtains He initialisation (He et al. 2015), that $\sigma_w = \sqrt{2}$, where $\Sigma^{1/2} = \mathrm{diag}(\sigma_w, ..., \sigma_w, 0)^\top$.

The analogue for GELU is more involved since no *single* σ_w preserves the expected square norms of *all* inputs. Setting $\sigma_b = 0$, $k(\mathbf{x}, \mathbf{x})/\sigma_w^2 = \|\mathbf{x}\|^2$ and $s_1 = s_2 = \sigma_w\|\mathbf{x}\|$ in the equation in Figure 2, we find a root $\sigma^*(\|\mathbf{x}\|)$ of

$$g_{\|\mathbf{x}\|}(\sigma) = \frac{\sigma^4\|\mathbf{x}\|^2}{\pi(\sigma^2\|\mathbf{x}\|^2 + 1)\sqrt{2\sigma^2\|\mathbf{x}\|^2 + 1}} + \frac{\sigma^2}{4}\left(1 + \frac{2}{\pi}\sin^{-1}\frac{\sigma^2\|\mathbf{x}\|^2}{1 + \sigma^2\|\mathbf{x}\|^2}\right) - 1,$$

numerically. Figure 4 shows a plot of $\sigma^*(\|\mathbf{x}\|)$ as $\|\mathbf{x}\|$ varies. The root of the limit of $g_{\|\mathbf{x}\|}(\sigma)$ as $\|\mathbf{x}\| \to \infty$ is $\sqrt{2}$, which recovers He initialisation. This implies that when data has large norms (such as images or audio files), He initialisation is suitable. The same procedure can be carried out for the ELU kernel as shown in Figure 4. This procedure may be viewed as a warm-up handling the special case of $\mathbf{x}_1 = \mathbf{x}_2$ for our general fixed point analysis.

4.2 General Fixed Point Analysis

Let $\mathcal{S} \subseteq [0,\infty) \times [0,\infty) \times [-1,1]$. In the infinitely wide limit, we may view each layer as updating a state $(s_1^2, s_2^2, \cos\theta) \in \mathcal{S}$ containing the expected square norms and the cosine angle between the signals in the hidden layers through a function $\mathbf{g} : \mathcal{S} \to \mathcal{S}$. Let $(G_1, G_2)^\top \sim \mathcal{N}(\mathbf{0}, I)$. We study the fixed-point dynamics of the iterated map $\mathbf{g}$ having components

$$\begin{aligned} &g_1(s_1^2, s_2^2, \rho) = \sigma_w^2 \mathbb{E}\big[\psi^2(s_1 G_1)\big] + \sigma_b^2, \\ &g_2(s_1^2, s_2^2, \rho) = \sigma_w^2 \mathbb{E}\big[\psi^2(s_2 G_2)\big] + \sigma_b^2 \\ &g_3(s_1^2, s_2^2, \rho) = \\ &\frac{\mathbb{E}\Big[\sigma_w^2 \psi(s_1 G_1)\psi\big(s_2(G_1\rho + G_2\sqrt{1-\rho^2})\big) + \sigma_b^2\Big]}{\sqrt{g_1(s_1^2, s_2^2, \rho) g_2(s_1^2, s_2^2, \rho)}}, \end{aligned} \tag{4}$$

which track the expected square norms (after a linear transformation involving σ_w^2 and σ_b^2) and normalised kernel as the signals propogate through the layers[3]. By inspection, g_3 (but not necessarily $\mathbf{g}$) always has an uncountable set of fixed points at $\rho = 1$ along $s_1 = s_2$. Banach's fixed point theorem says that if $\mathbf{g}$ is a contraction mapping on a closed set, then $\mathbf{g}$ has a unique fixed point on that set (Agarwal, Meehan, and O'regan 2001). In a slightly different setting, Hasselblatt and Katok (2003, Theorem 2.2.16) allows some open sets.

Theorem 5. *Let $D\mathbf{g}$ denote the Jacobian of $\mathbf{g}$ and d' denote the metric induced by a norm with induced matrix norm $\|\cdot\|'$. If $C \subset \mathbb{R}^m$ is an open strictly convex set, $\overline{C}$ is its closure, $\mathbf{g} : \overline{C} \to \overline{C}$ differentiable on C and continuous on $\overline{C}$ with $\|D\mathbf{g}\|' \leq \lambda < 1$ on C, then $\mathbf{g}$ has a unique fixed point $\mathbf{c}_0 \in \overline{C}$ and $d'\big(\mathbf{g}^L(\mathbf{c}), \mathbf{c}_0\big) \leq \lambda^L d'(\mathbf{c}, \mathbf{c}_0)$ for every $\mathbf{c} \in \overline{C}$.*

We therefore consider the eigenvalues of the Jacobian, proving the following in Appendix C.

Theorem 6. *Let $\mathbf{g}$ be as in (4), and suppose the absolute value of ψ is bounded by a polynomial. Let $(Z_1, Z_2) \sim \mathcal{N}(\mathbf{0}, S)$ with covariance $\rho = \cos\theta$ and unit variances. Then for $\rho \in (-1,1)$ $0 < s_1, s_2$, the (unordered) eigenvalues of the Jacobian of $\mathbf{g}$ are*

$$\begin{aligned} \lambda_1 &:= \frac{\partial g_1}{\partial s_1^2} = \sigma_w^2 \mathbb{E}\Big[\big(Z_1^2 - 1\big)\psi^2(s_1 Z_1)\Big]/(2s_1^2), \\ \lambda_2 &:= \frac{\partial g_2}{\partial s_2^2} = \sigma_w^2 \mathbb{E}\Big[\big(Z_2^2 - 1\big)\psi^2(s_2 Z_2)\Big]/(2s_2^2), \quad \textit{and} \\ \lambda_3 &:= \frac{\partial g_3}{\partial \rho} = \frac{\sigma_w^2 s_1 s_2}{\sqrt{g_1 g_2}} \mathbb{E}\big[\psi'(s_1 Z_1)\psi'(s_2 Z_2)\big], \end{aligned}$$

provided the right hand terms are finite, where ψ' is the distributional derivative of ψ.

Our result does not include $\rho \in \{-1, 1\}$. With the additional assumption that ψ is continuous almost everywhere, the expression for λ_3 is valid on the closed interval $\rho \in [-1, 1]$, as shown in Appendix F. We would now like to combine Theorem 5 and Theorem 6 in order to comment on the existence of unique fixed points in some special cases. We consider two general cases in Corollaries 7 and 8 below.

Corollary 7 (Unique fixed point under absolute homogeneity)**.** *Suppose $\sigma_b^2 = 0$ (as is common when initialising neural networks) and ψ is absolutely homogeneous, that is, $\psi(|a|z) = |a|\psi(z)$ for any $a \in \mathbb{R}$. Then*

$$\frac{\partial g_3}{\partial \rho} = \lambda_3 = \frac{\mathbb{E}\big[\psi'(Z_1)\psi'(Z_2)\big]}{\mathbb{E}[\psi^2(Z_1)]}.$$

Furthermore, if

- $\max_{i=1,2,3} |\lambda_i| < 1$ *then* $\mathbf{g}$ *admits a unique fixed point at* $\big(s^2 s^2, 1\big)$ *for some* s^2.
- $\lambda_3 < 1$ *and* $g_1(\cdot, s_2^2, \rho)$ *admits a fixed point* s^2 *for any* s_2^2, ρ, *then* $g_3(s^2, s^2, \cdot)$ *admits a unique fixed point at* 1.

Proof. Absolute homogeneity implies that

$$\begin{aligned} &g_3(s_1^2, s_2^2, \rho) \\ &= \frac{\mathbb{E}\Big[\sigma_w^2 \psi(s_1 G_1)\psi\big(s_2(G_1\rho + G_2\sqrt{1-\rho^2})\big)\Big]}{\sqrt{\mathbb{E}\Big[\sigma_w^2\psi^2(s_1 G_1)\big)\Big]}\sqrt{\mathbb{E}\Big[\sigma_w^2\psi^2(s_2 G_2)\big)\Big]}} \\ &= \frac{\mathbb{E}\Big[\psi(G_1)\psi\big(G_1\rho + G_2\sqrt{1-\rho^2}\big)\Big]}{\sqrt{\mathbb{E}\Big[\psi^2(G_1)\big)\Big]}\sqrt{\mathbb{E}\Big[\psi^2(G_2)\big)\Big]}} \\ &= 0 = \frac{\partial g_3}{\partial s_1^2} = \frac{\partial g_3}{\partial s_2^2}. \end{aligned}$$

Absolute homogeneity also implies that for all $a \in \mathbb{R}$, $\psi'(|a|z) = \psi'(z)$. Then by Theorem 6,

$$\begin{aligned} \lambda_3 &= \frac{\sigma_w^2 s_1 s_2}{\sqrt{g_1 g_2}} \mathbb{E}\big[\psi'(s_1 Z_1)\psi'(s_2 Z_2)\big] \\ &= \frac{\sigma_w^2 s_1 s_2}{\sigma_w^2 s_1 s_2 \mathbb{E}\big[\psi^2(G_1)\big]} \mathbb{E}\big[\psi'(Z_1)\psi'(Z_2)\big] \\ &= \frac{\mathbb{E}\big[\psi'(Z_1)\psi'(Z_2)\big]}{\mathbb{E}\big[\psi^2(G_1)\big]}. \end{aligned}$$

Note that the Jacobian

$$\begin{pmatrix} \frac{\partial g_1}{\partial s_1^2} & \frac{\partial g_1}{\partial s_2^2} & \frac{\partial g_1}{\partial \rho} \\ \frac{\partial g_2}{\partial s_1^2} & \frac{\partial g_2}{\partial s_2^2} & \frac{\partial g_2}{\partial \rho} \\ \frac{\partial g_3}{\partial s_1^2} & \frac{\partial g_3}{\partial s_2^2} & \frac{\partial g_3}{\partial \rho} \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

is diagonal, and therefore the induced matrix norm (corresponding to a Euclidean vector norm) of the Jacobian, the largest singular value of the Jacobian, is simply the largest

[3] We expressed the kernel (1) in terms of iid Gaussians $\mathbf{G}$ instead of dependent Gaussians $\mathbf{Z}$.

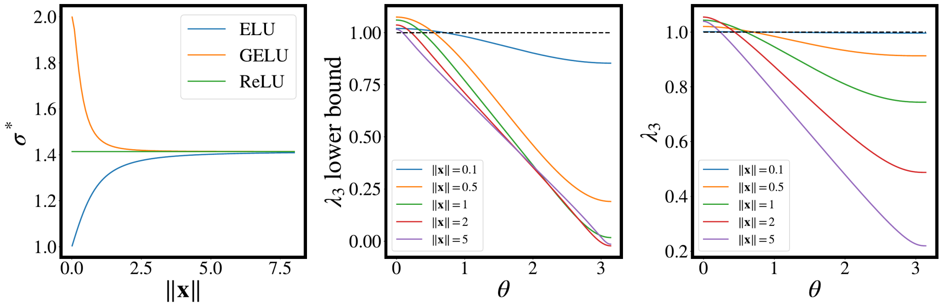


Figure 4: (Left) Values σ^* that preserve the layer-wise expected square norm of $\|\mathbf{x}\|$ in MLPs with GELU, ELU and ReLU activations. (Middle) lower bound of λ_3 for GELU (Right) λ_3 for ELU. If $\lambda_3 \leq 1$ on $\theta \in (0, \pi)$, a unique fixed point exists.

absolute value of the diagonal elements. Thus by Theorem 5, if $\max_{i=1,2,3} |\lambda_i| < 1$ then $\mathbf{g}$ has a unique fixed point.

Alternatively, suppose $\lambda_3 < 1$ and $g_1(\cdot, s_2^2, \rho)$ admits a fixed point at s^2 for any s_2^2, ρ. Then $g_3\big(s^2, s^2, \cdot\big)$ admits a unique fixed point by applying Theorem 5 to the 1D system g_3 with induced matrix norm $|\lambda_3|$. □

Intuitively but informally, the absolute homogeneity condition in Corollary 7 leads to independent updates, so that $\mathbf{g}$ may be thought of as three functions g_1, g_2, g_3 whose inputs and outputs do not interact between iterations. When absolute homogeneity is removed, g_1 and g_2's inputs and outputs are not affected by g_3, but g_3's inputs are affected by the outputs of g_1 and g_2. This makes it more difficult to analyse exactly the same situation as in Corollary 7. However, if we fix the output of g_1 and g_2 at some fixed point (which are guaranteed to exist in the cases handled in § 4.1), then we can neglect interactions between the outputs of g_1, g_2 and the inputs of g_3 and analyse the iterates of g_3 as a univariate function. We proceed with this strategy in Corollary 8.

Corollary 8 (Unique fixed point of normalised kernel). *If a fixed point s^2 of the system only involving $g_1(\cdot, s_2^2, \rho)$: $[0, \infty) \to [0, \infty)$ exists when $\sigma_w = \sigma^*$, we have*

$$\frac{\partial g_3}{\partial \rho} = \lambda_3 = (\sigma^*)^2 \mathbb{E}\big[\psi'(sZ_1)\psi'(sZ_2)\big]$$

at the fixed point of $g_1(\cdot, s_2^2, \rho)$ and $g_2(s_1^2, \cdot, \rho)$. Furthermore, if $|\lambda_3| < 1$, then $g_3(s^2, s^2, \cdot) : [-1, 1] \to [-1, 1]$ admits a unique fixed point at $\rho = 1$.

Proof. By Theorem 6, we have

$$\begin{aligned}\lambda_3 &= \frac{\sigma_w^2 s_1 s_2}{\sqrt{g_1 g_2}} \mathbb{E}\big[\psi'(s_1 Z_1)\psi'(s_2 Z_2)\big] \\ &= (\sigma^*)^2 \mathbb{E}\big[\psi'(sZ_1)\psi'(sZ_2)\big].\end{aligned}$$

$g_3(s^2, s^2, \cdot)$ admits a unique fixed point by taking $C = (-1, 1)$ and d as the the Euclidean metric in Theorem 5. □

4.3 Degenerate Priors and Posteriors

Having established conditions under which unique fixed points exist, we now examine what a unique fixed point implies for the limiting prior and posterior. The prior and posteriors are degenerate in the sense that they are almost surely constant over subsets of the input space. We first consider the limiting prior as the depth goes to infinity.

Proposition 9. *Let $\{f^{(L)}(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$ be a Gaussian process with mean zero and covariance function $k^{(L)}$. Suppose that $\lim_{L\to\infty} k^{(L)}(\mathbf{x}_1, \mathbf{x}_2) = \lim_{L\to\infty} k^{(L)}(\mathbf{x}_1, \mathbf{x}_1) = \lim_{L\to\infty} k^{(L)}(\mathbf{x}_2, \mathbf{x}_2) < \infty$ for every $\mathbf{x}_1$, $\mathbf{x}_2 \in \mathcal{X}_*$. Then*

$$\lim_{L\to\infty} f^{(L)}(\mathbf{x}_1) - f^{(L)}(\mathbf{x}_2) = 0$$

almost surely. That is, all draws from the limiting prior are almost surely a constant function.

The proof is given in Appendix H. Suppose we take *a priori* the Gaussian process in Proposition 9 *before the limit is taken*, update our belief after observing some data to obtain the posterior *and then* take the limit. An interesting question is whether the limit commutes with the Bayesian update.

Proposition 10. *Let $\{f^{(L)}(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$ be a Gaussian process prior with mean zero and covariance function $k^{(L)}$. Fix some $\mathcal{X}_* \subseteq \mathcal{X}$ such that for every $\mathbf{x}_1$, $\mathbf{x}_2 \in \mathcal{X}_* \subseteq \mathcal{X}$ and $\mathbf{x}_3, \mathbf{x}_4 \in \mathcal{X}$,*

- $\lim_{L\to\infty} k^{(L)}(\mathbf{x}_1, \mathbf{x}_2) = \lim_{L\to\infty} k^{(L)}(\mathbf{x}_1, \mathbf{x}_1) = \lim_{L\to\infty} k^{(L)}(\mathbf{x}_2, \mathbf{x}_2) = C < \infty$,
- $\lim_{L\to\infty} k^{(L)}(\mathbf{x}_1, \mathbf{x}_3) = \lim_{L\to\infty} k^{(L)}(\mathbf{x}_2, \mathbf{x}_3) = D(\mathbf{x}_3)$*, and*
- $\lim_{L\to\infty} k^{(L)}(\mathbf{x}_3, \mathbf{x}_4) < \infty$ *exists*

where $C \in \mathbb{R}$ and $D : \mathcal{X} \to \mathbb{R}$ may depend on $\mathcal{X}_*$. Fix some dataset $\mathbf{X}, \mathbf{Y}$, where each row $\mathbf{X}_i$ of $\mathbf{X}$ is in $\mathcal{X}$.

Then under Bayesian Gaussian process regression with Gaussian likelihood and strictly positive noise variance $\sigma_n^2 > 0$, all draws from the limiting posterior predictive distribution given observations $\mathbf{X}, \mathbf{Y}$ over $\mathcal{X}_*$ as $L \to \infty$ are almost surely a constant function.

The proof is given in Appendix H. We are now ready to relate the existence of unique fixed points to degenerate priors and posteriors for some specific examples.

Example, LReLU Taking $\mathcal{X}_*$ to be any (subset of a) hypersphere in Propositions 9 and 10, we have the following result, the proof of which is given in Appendix H.

Corollary 11. *Let $\mathcal{X}_*$ be any hypersphere. Define a Gaussian process prior with a covariance function corresponding to an infinitely wide MLP with LReLU activations, $\boldsymbol{\mu} = \mathbf{0}$, $\sigma_w^2 = 2$ and depth L. Then as $L \to \infty$, draws from the prior and posterior predictive distributions are almost surely constant over $\mathcal{X}_*$.*

Examples, GELU and ELU In contrast with LReLU activations, such a degeneracy guarantee does not exist for GELU and ELU activations. For both the GELU and ELU, we consider the dynamics on a ball of constant $\|\mathbf{x}\|$, where σ_w is chosen such that $g_1 = \|\mathbf{x}\|$. In Figure 4, for different values of $\|\mathbf{x}\|$ in the context of Corollary 8, we evaluate a lower bound for λ_3 in the case of GELU and λ_3 exactly in the case of ELU. Full working is given in Appendices G.1 and G.2. We observe that each exceeds 1 *at some point on the* (but not over the whole) interval, and is therefore not a contraction mapping and hence not guaranteed to have a unique fixed point. This is consistent with Figures 2 and 3, where fixed points are shown by intersecting curves.

5 Extension of Theoretical Results to NTK

We may also study kernel fixed points of infinitely wide neural networks trained under gradient flow. This amounts to studying the fixed point properties of the neural tangent kernel (NTK). If such a unique fixed point exists, this implies that the functions obtained by applying gradient flow to an infinitely wide, infinitely deep MLP are also degenerate. In this section, we briefly sketch how such a result may be obtained. We leave the presentation of formal results and empirical evaluations for future work.

Informally, the value of the eigenvalue λ_3 can still predict the fixed point behaviour of the NTK. Formally, the result is slightly more involved, see Appendix I. Consider a state space $\mathcal{S} \subset \mathbb{R}^4$ containing states (s_1^2, s_2^2, k, T) consisting of squared norms for both inputs, the kernel, and the NTK. We update the states through $\mathbf{h} : \mathcal{S} \to \mathcal{S}$:

$$\begin{aligned} h_i(s_1^2, s_2^2, k, T) &= \sigma_w^2 \mathbb{E}\big[\psi^2(s_i Z_i)\big] + \sigma_b^2, \quad i = i, 2 \\ h_3(s_1^2, s_2^2, k, T) &= \sigma_w^2 \mathbb{E}\big[\psi(s_1 Z_1)\psi(s_2 Z_2)\big] + \sigma_b^2, \\ h_4(s_1^2, s_2^2, k, T) &= T\sigma_w^2 \mathbb{E}\big[\psi'(s_1 Z_1)\psi'(s_2 Z_2)\big] + \\ &\quad \sigma_w^2 \mathbb{E}\big[\psi(s_1 Z_1)\psi(s_2 Z_2)\big] + \sigma_b^2, \end{aligned}$$

where $\mathrm{Cov}(Z_1, Z_2) = k/(s_1 s_2)$, $\mathbb{E}[Z_1] = \mathbb{E}[Z_2] = 0$. As in Corollary 8, if a fixed point of the system only involving s_1 and h_1 exists at a value $\sigma_w = \sigma^*$ and $s_1 = s_2 = s$, then the system reduces to a 2 dimensional update involving only h_3 and h_4 along $s_1 = s_2 = s$. The Jacobian is diagonal,

$$J = \begin{pmatrix} \frac{\partial h_3}{\partial k} & \frac{\partial h_3}{\partial T} \\ \frac{\partial h_4}{\partial k} & \frac{\partial h_4}{\partial T} \end{pmatrix} = \begin{pmatrix} \frac{\partial h_3}{\partial k} & 0 \\ \frac{\partial h_4}{\partial k} & \frac{\partial h_4}{\partial T} \end{pmatrix},$$

so the eigenvalues are $\frac{\partial h_3}{\partial k}$ and $\frac{\partial h_4}{\partial T}$. By Theorem 6,

$$\begin{aligned} \frac{\partial h_3}{\partial k} &= \frac{\partial h_3}{\partial \rho}\Big(\frac{\partial k}{\partial \rho}\Big)^{-1} = \frac{\partial g_3}{\partial \rho}\sqrt{h_1 h_2}(s_1 s_2)^{-1} \\ &= (\sigma^*)^2 \mathbb{E}\big[\psi'(s_1 Z_1)\psi'(s_2 Z_2)\big] = \frac{\partial h_4}{\partial T} = \lambda_3. \end{aligned}$$

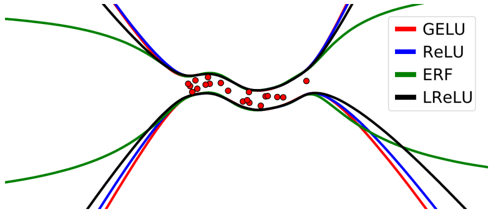


Figure 5: Posterior predictive ±2 standard deviations.

6 Gaussian Process Experiments

We perform two sets of experiments. In the first, we investigate the performance of GPs with various neural network kernels. In the second, we observe the degree of implicit regularisation that is obtained using GPs of finite depth.

6.1 Benchmarking

We provide a software implementation of our new kernels. To demonstrate usage of our covariance functions, first we compare the performance of GP regression models using ReLU, LReLU, ERF and GELU kernels on a popular Bayesian deep learning benchmark (Hernández-Lobato and Adams 2015). The purpose of these experiments is not to showcase the superiority of one prior over another, but rather to provide a sample implementation. This implementation has already been ported over to another framework in concurrent work by others (Novak et al. 2020). The ELU kernel was not included in our experiments (see § 3). We perform separate experiments on shallow models having 1 hidden layer and deep models having up to 32 hidden layers. All data was standardised to have mean 0 and variance 1.

Shallow models. *Do differences in priors induced by the various activation functions affect empirical performance?* Using the limiting GP allows us to remove the interaction between ψ and optimisation, and purely consider the effect of ψ on the functional prior. Figure 5 shows the predictive distribution of GPs with GELU, ReLU, LReLU and ERF kernels on a toy regression task. ERF has different extrapolation properties due to being a bounded activation, whilst the others appear qualitatively similar, though with extrapolation variance decreasing in the order GELU/ReLU/LReLU.

Figure 6 shows benchmark results for single-hidden-layer GPs using a 90%/10% training/test split. See Appendix J.1 for more details and plots. All kernels perform comparably; gains can be made by selecting a kernel suited to the dataset. Results are most different for ERF — either negatively (Concrete, Energy) or positively (Boston, Protein, Wine). Differences are observed between GELU/ReLU/LReLU. For example, GELU offers an advantage in Naval and Yacht, and LReLU performs poorly on Protein.

None of the kernels consistently outperform the others. This is expected behaviour, similar to how a Matern kernel might outperform a squared exponential kernel only some of the time on real-world datasets. The purpose of the experiment was to evaluate whether different ψ result in a strong enough difference in priors that empirical performance differences can be observed. Having answered in the positive,

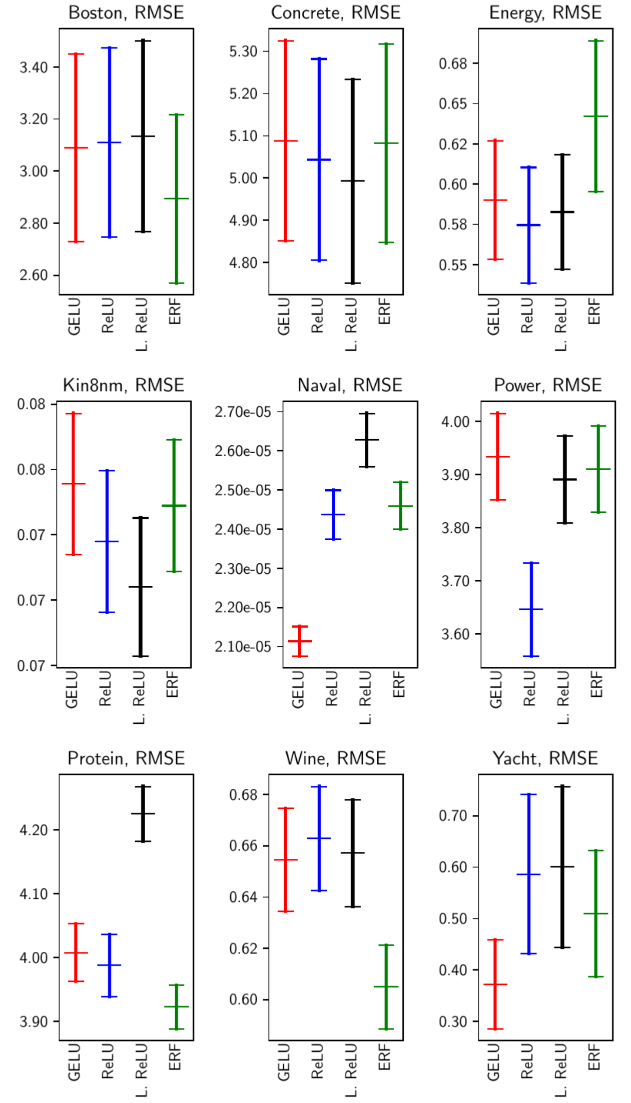


Figure 6: RMSE for equivalent single-hidden-layer GPs. Mean ±2 standard errors (over 20 runs).

we posit that for finite-width networks, the difference in performance found by varying ψ may partially derive from differences in the induced prior. This is in contrast to previously cited reasons such as bias shift and its relation to natural gradient (Clevert, Unterthiner, and Hochreiter 2016).

Deep models. *How does the performance of models vary with depth?* We randomly shuffled the data into an $80/20\%$ train/test split 5 times. For each split, we ran GP regression with an additive iid Gaussian noise model having variance fixed at 0.1. We varied the depth $\ell \in [1, 32]$ in steps of 1 and the weight and bias variances (which were constrained to be equal in each layer) $\sigma_w^2 \in [0.1, 5]$ in steps of 0.1. For each setting, we measured the RMSE between the mean of the GP prediction and the true regression targets. Figure 7 shows the average RMSE on the Wine dataset over 5 shuffles. Other datasets are given in Appendix J.2. We make two qualitative observations. Firstly, $\ell > 1$ models out-perform $\ell = 1$ models. Secondly, the RMSE changes smoothly in both depth and σ_w^2. The visual smoothness is *not* due to averaging over 5 trials; smoothness is also observed when we plot results from only 1 random shuffling. Table 1 shows the best models obtained over the grid search for each kernel.

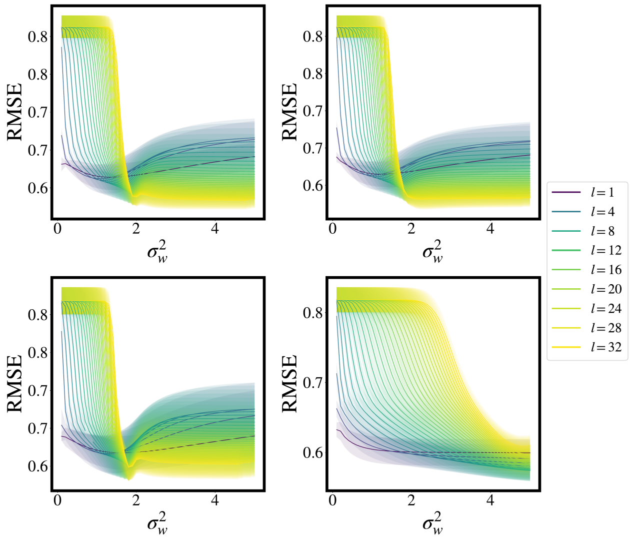


Figure 7: RMSE against σ_w^2 for equivalent l layer GPs, Wine dataset. Shaded region shows ±1 standard deviation. (Clockwise from top left) ReLU, GELU, LReLU, ERF.

6.2 Overfitting and Underfitting

We empirically investigate the relationship between depth and training/testing error. When the covariance function has a unique fixed point, we expect to see underfitting at large depth, since large depth will push the kernel towards the unique fixed point, that is, a constant normalised kernel. On the other hand, when the covariance function does not have a unique fixed point, we might expect to see overfitting as model complexity may increase with depth.

We build predictor variables of a training dataset by uniformly sampling $\mathbf{x} \in \mathbb{R}^2$ on the unit disc at heading γ. We then sample training targets through the mapping $y = f(\gamma) + \epsilon$ for a number of different choices of f, where ϵ is additive Gaussian noise with variance fixed at 0.1. We find the posterior predictive mean of a Gaussian process with iterated GELU and ReLU covariance functions of depth L between 1 and 100 with a choice of σ_w according to Figure 4. We repeat this process with a new random training set 10 times. Each repetition, we find the mean-squared error (MSE) between the posterior mean of the Gaussian process over the training set and a test set built from a (deterministic) uniform grid of size 100. Figure 8 shows the resulting train and test errors on one choice of f, and Appendix L shows other choices of f. Figure 1 shows a more direct illustration of the function fit on one of the random data samples, with other choices of f shown in Appendix L.

	ReLU			GELU			LReLU			ERF		
	RMSE	σ_w^2	ℓ	RMSE	σ_w^2	ℓ	RMSE	σ_w^2	ℓ	RMSE	σ_w^2	ℓ
Boston	2.85 ± 0.64	1.90	7	2.86 ± 0.65	1.80	6	$\mathbf{2.60 \pm 1.07}$	$\mathbf{2.00}$	$\mathbf{32}$	2.69 ± 0.95	5.00	2
Concrete	5.22 ± 0.55	5.00	2	$\mathbf{5.21 \pm 0.56}$	$\mathbf{5.00}$	$\mathbf{2}$	5.23 ± 0.44	3.30	3	5.63 ± 0.46	5.00	2
Energy	$\mathbf{0.89 \pm 0.11}$	$\mathbf{5.00}$	$\mathbf{2}$	0.92 ± 0.12	5.00	2	2.77 ± 0.31	0.10	1	2.79 ± 0.23	0.10	1
Wine	1.15 ± 0.13	5.00	4	1.17 ± 0.14	5.00	4	$\mathbf{1.04 \pm 0.12}$	$\mathbf{5.00}$	$\mathbf{5}$	3.96 ± 0.75	5.00	1
Yacht	0.58 ± 0.01	4.80	32	0.58 ± 0.01	2.30	32	0.60 ± 0.02	1.80	29	$\mathbf{0.57 \pm 0.02}$	$\mathbf{5.00}$	$\mathbf{8}$

Table 1: Best performing models for each kernel over the grid search.

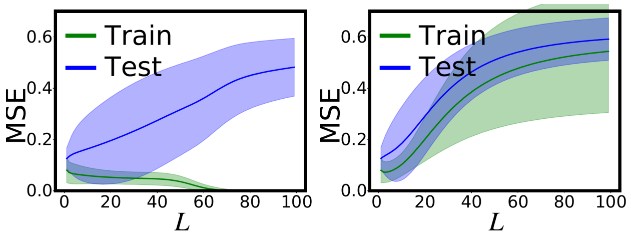


Figure 8: Training and testing errors for GPs with covariance functions corresponding to infinitely wide MLPs of increasing depth L using the same example as in Figure 1. Solid curve shows the mean over 10 training data samples, and the shaded region shows ± two standard deviations. By examining whether the training and testing error increases or decreases with depth, we observe that the GELU and ReLU respectively overfit and underfit with depth. See Appendix L for more error curves. (Left) GELU (Right) ReLU.

7 Discussion and Conclusion

We introduced two new positive semi-definite kernels arising from the infinite-width limit of Bayesian GELU or ELU NNs. We provided visualisations of these kernels for varying depths. We introduced a general framework for understanding the fixed-point dynamics of such kernels and their NTK counterparts. Using this framework, we showed that unlike the ReLU, the GELU and ELU kernels are able to avoid unique fixed points. We empirically verified that finite-width NNs are able to avoid unique kernel fixed points in Figures 2 and 3. We applied our kernels in the setting of shallow and deep GP regression, finding that for some problems specific kernels are more appropriate, and that the GELU kernel is competitive with the ReLU kernel.

Investigations into implicit regularisation consider the role of one or all of (a) the architecture, (b) the learning algorithm and (c) the data sampling process. Neyshabur, Tomioka, and Srebro (2015) argue that (b) leads implicitly to low-norm solutions, explaining the generalisation ability of deep NNs. On the other hand, Dereziński, Liang, and Mahoney (2019) construct a data sampling distribution (c) that explains double descent and implicit regularisation in linear models. Similar to our work but not considering the NTK, the signal propagation literature (Schoenholz et al. 2017; Poole et al. 2016) explains simplicity biases in randomly initialised networks (a). They develop objects similar to $\frac{\partial g_3}{\partial \rho}$, but require bounded activations, and seem to also require some notion of differentiability. Our analysis considers (a) and (b) but not (c). We have recently been made aware of the concurrent work of (Huang et al. 2020), who also study the degeneracy of processes induced through ReLU activations. Their focus is on both (a) and (b), arguing that the NTK for residual architectures does not suffer from this degeneracy.

Knowing the gradient of the kernel with respect to $(\sigma_w^{(l)})^2$ and $(\sigma_b^{(l)})^2$ is useful for both empirical Bayesian methodologies (e.g. optimising the marginal likelihood using LBFGS) and hierarchical models (e.g. using HMC to integrate out the hyperprior). As we detail in Appendix K, our results may be used to find this gradient. Theorem 6 provides 7 of the 9 elements of the Jacobian. The other 2 can only easily be evaluated in special cases (e.g. LReLU results in a diagonal Jacobian). In future work, it may be interesting to extend Theorem 6 to cover the remaining elements of the Jacobian.

While Lee et al. (2019) found close agreement between NNs and their corresponding limiting GPs, several authors (Neal 1995; MacKay 2003; Der and Lee 2006; Matthews et al. 2018; Chizat, Oyallon, and Bach 2019; Tsuchida, Roosta, and Gallagher 2019b; Allen-Zhu, Li, and Liang 2019; Peluchetti, Favaro, and Fortini 2020; Aitchison 2020) have argued against the use of GP models as a means to understand the success of deep learning. If deep learning's performance can be explained using GPs, why do NN models outperform their limiting GP counterparts? Arora et al. (2019) attain 77% test accuracy on CIFAR10 using a limiting GP arising from a trained CNN, while a ResNet is able to achieve 96% (Springenberg et al. 2015). On the other hand, Arora et al. (2020) find that GP models are competitive on *small* datasets. It remains to determine if this difference in performance is due to the tricks for which equivalence in the GP setting have not yet been fully explored, or if it is the result of some deeper property of GPs. Lee et al. (2020) empirically explore some of these questions including the effects of finite width, architectures, weight decay and the performance difference between infinite Bayesian and NTK models. While we acknowledge the limitations of the infinitely wide approach, we believe it warrants further exploration, if not to understand the power of deep learning, at least to investigate its generalisation abilities. The purpose of our study was not to optimise any architecture for performance on a particular problem, but rather to develop results under the GP framework that contribute to our understanding of generalisation in the overparameterised setting.

Acknowledgements

This work was partially funded by CSIRO's Machine Learning and Artificial Intelligence Future Science Platform. TP was funded through EPSRC (EP/N509620/1). CvdH was supported by ACEMS under ARC grant number CE140100049. We would like to thank Bob Williamson for a helpful discussion.

References

Agarwal, R. P.; Meehan, M.; and O'regan, D. 2001. *Fixed point theory and applications*, volume 141. Cambridge university press.

Aitchison, L. 2020. Why bigger is not always better: on finite and infinite neural networks. In *International Conference on Machine Learning*, 156–164.

Allen-Zhu, Z.; Li, Y.; and Liang, Y. 2019. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, 6155–6166.

Arora, S.; Du, S. S.; Hu, W.; Li, Z.; Salakhutdinov, R. R.; and Wang, R. 2019. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, 8139–8148.

Arora, S.; Du, S. S.; Li, Z.; Salakhutdinov, R.; Wang, R.; and Yu, D. 2020. Harnessing the Power of Infinitely Wide Deep Nets on Small-data Tasks. In *International Conference on Learning Representations*.

Billingsley, P. 1995. *Probability and measure*. John Wiley & Sons, 3rd edition.

Bui, T.; Hernández-Lobato, D.; Hernandez-Lobato, J.; Li, Y.; and Turner, R. 2016. Deep Gaussian processes for regression using approximate expectation propagation. In *International conference on machine learning*, 1472–1481.

Chizat, L.; Oyallon, E.; and Bach, F. 2019. On Lazy Training in Differentiable Programming. In *Advances in Neural Information Processing Systems*.

Cho, Y.; and Saul, L. K. 2009. Kernel Methods for Deep Learning. In *Advances in Neural Information Processing Systems*, 342–350.

Chua, K.; Calandra, R.; McAllister, R.; and Levine, S. 2018. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, 4754–4765.

Clevert, D.-A.; Unterthiner, T.; and Hochreiter, S. 2016. Fast and accurate deep network learning by exponential linear units (ELUs). In *The International Conference on Learning Representations*.

Der, R.; and Lee, D. D. 2006. Beyond Gaussian processes: On the distributions of infinite networks. In *Advances in Neural Information Processing Systems*, 275–282.

Dereziński, M.; Liang, F.; and Mahoney, M. W. 2019. Exact expressions for double descent and implicit regularization via surrogate random design. In *arXiv preprint arXiv:1912.04533*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Elfwing, S.; Uchibe, E.; and Doya, K. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks* 107: 3–11.

Garriga-Alonso, A.; Rasmussen, C. E.; and Aitchison, L. 2018. Deep Convolutional Networks as shallow Gaussian Processes. In *International Conference on Learning Representations*.

Hasselblatt, B.; and Katok, A. 2003. *A first course in dynamics: with a panorama of recent developments*. Cambridge University Press.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Hendrycks, D.; and Gimpel, K. 2016. Gaussian error linear units (GELUs). In *arXiv preprint arXiv:1606.08415*.

Hernández-Lobato, J. M.; and Adams, R. P. 2015. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In *Proceedings of the 32nd International Conference on Machine Learning*.

Huang, K.; Wang, Y.; Tao, M.; and Zhao, T. 2020. Why Do Deep Residual Networks Generalize Better than Deep Feedforward Networks?–A Neural Tangent Kernel Perspective. *Advances in Neural Information Processing Systems* 33.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, 8571–8580.

Jones, D. S. 1982. *The theory of generalised functions*. Cambridge ; New York: Cambridge University Press, 2nd ed. edition.

Klambauer, G.; Unterthiner, T.; Mayr, A.; and Hochreiter, S. 2017. Self-normalizing neural networks. In *Advances in neural information processing systems*, 971–980.

Le Roux, N.; and Bengio, Y. 2007. Continuous neural networks. In *Artificial Intelligence and Statistics*, 404–411.

Lee, J.; Bahri, Y.; Novak, R.; Schoenholz, S. S.; Pennington, J.; and Sohl-Dickstein, J. 2018. Deep neural networks as Gaussian processes. In *The International Conference on Learning Representations*.

Lee, J.; Schoenholz, S.; Pennington, J.; Adlam, B.; Xiao, L.; Novak, R.; and Sohl-Dickstein, J. 2020. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems* 33.

Lee, J.; Xiao, L.; Schoenholz, S.; Bahri, Y.; Novak, R.; Sohl-Dickstein, J.; and Pennington, J. 2019. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, 8570–8581.

MacKay, D. J. 2003. *Information theory, inference and learning algorithms*, 547. Cambridge university press.

Matthews, A. G. d. G.; Rowland, M.; Hron, J.; Turner, R. E.; and Ghahramani, Z. 2018. Gaussian process behaviour in wide deep neural networks. In *The International Conference on Learning Representations*.

Meronen, L.; Irwanto, C.; and Solin, A. 2020. Stationary Activations for Uncertainty Calibration in Deep Learning. *Advances in Neural Information Processing Systems* 33.

Neal, R. M. 1995. *Bayesian learning for neural networks*. Ph.D. thesis, University of Toronto.

Neyshabur, B.; Tomioka, R.; and Srebro, N. 2015. In search of the real inductive bias: On the role of implicit regularization in deep learning. In *International Conference on Learning Representations (workshop track)*.

Novak, R.; Xiao, L.; Hron, J.; Lee, J.; Alemi, A. A.; Sohl-Dickstein, J.; and Schoenholz, S. S. 2020. Neural Tangents: Fast and Easy Infinite Neural Networks in Python. In *International Conference on Learning Representations*. URL https://github.com/google/neural-tangents.

Novak, R.; Xiao, L.; Lee, J.; Bahri, Y.; Yang, G.; Hron, J.; Abolafia, D. A.; Pennington, J.; and Sohl-Dickstein, J. 2019. Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes. In *The International Conference on Learning Representations*.

Pearce, T.; Tsuchida, R.; Zaki, M.; Brintrup, A.; and Neely, A. 2019. Expressive Priors in Bayesian Neural Networks: Kernel Combinations and Periodic Functions. In *Uncertainty in Artificial Intelligence*.

Peluchetti, S.; Favaro, S.; and Fortini, S. 2020. Stable behaviour of infinitely wide deep neural networks. In *International Conference on Artificial Intelligence and Statistics*, 1137–1146. PMLR.

Poole, B.; Lahiri, S.; Raghu, M.; Sohl-Dickstein, J.; and Ganguli, S. 2016. Exponential expressivity in deep neural networks through transient chaos. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*, 3360–3368.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. In *Preprint*.

Ramachandran, P.; Zoph, B.; and Le, Q. V. 2017. Searching for activation functions. In *arXiv preprint arXiv:1710.05941*.

Rosenbaum, S. 1961. Moments of a truncated bivariate normal distribution. *Journal of the Royal Statistical Society: Series B (Methodological)* 23(2): 405–408.

Schoenholz, S. S.; Gilmer, J.; Ganguli, S.; and Sohl-Dickstein, J. 2017. Deep information propagation. In *International Conference on Learning Representations*.

Sheppard, W. F. 1899. On the application of the theory of error to cases of normal distribution and normal correlation. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* (192): 140.

Springenberg, J.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2015. Striving for Simplicity: The All Convolutional Net. In *International Conference on Learning Representations (workshop track)*.

Tsuchida, R.; Roosta, F.; and Gallagher, M. 2018. Invariance of Weight Distributions in Rectified MLPs. In *International Conference on Machine Learning*, 5002–5011.

Tsuchida, R.; Roosta, F.; and Gallagher, M. 2019a. Exchangeability and Kernel Invariance in Trained MLPs. In *International Joint Conference on Artificial Intelligence*.

Tsuchida, R.; Roosta, F.; and Gallagher, M. 2019b. Richer priors for infinitely wide multi-layer perceptrons. In *arXiv preprint arXiv:1911.12927*.

Valle-Pérez, G.; Camargo, C. Q.; and Louis, A. A. 2019. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *International Conference on Learning Representations*.

Williams, C. K. 1997. Computing with infinite networks. In *Advances in neural information processing systems*, 295–301.

Yang, G. 2019a. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. In *arXiv preprint arXiv:1902.04760*.

Yang, G. 2019b. Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes. In *Advances in Neural Information Processing Systems*, 9947–9960.

Yang, G.; and Salman, H. 2019. A fine-grained spectral perspective on neural networks. In *arXiv preprint arXiv:1907.10599*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*.