Meta-learning inductive biases of learning systems with Gaussian processes

Michael Y. Li
Princeton University
myli@alumni.princeton.edu

Erin Grant
UC Berkeley
eringrant@berkeley.edu

Thomas L. Griffiths
Princeton University
tomg@princeton.edu

Abstract

Many advances in machine learning can be attributed to designing systems with inductive biases well-suited for particular tasks. However, it can be challenging to ascertain the inductive biases of a learning system, much less control them in the design process. We propose a framework to capture the inductive biases in a learning system by meta-learning Gaussian process kernel hyperparameters from its predictions. We illustrate the potential of this framework across several case studies, including investigating the inductive biases of both untrained and trained neural networks, and assessing whether a given neural network family is well-suited for a task family.

1 Introduction

Many advances in machine learning can be attributed to the introduction of architectures with inductive biases well-suited for particular kinds of data. For example, the success of convolutional neural networks in computer vision [LeCun et al., 2015] is attributed to their architectural constraint of translational invariance, which is an appropriate bias for many of the problems to which they are applied. However, the inductive biases of a given machine learning system can often be unclear. While certain design decisions (*e.g.,* convolutional layers) render the inductive biases of a neural network explicit (*e.g.,* translational invariance), the inductive biases induced by many other design choices, such as the parameter initialization scheme, the architecture, and the training procedure, are less clear. While there have been both empirical [Linzen et al., 2016, McCoy et al., 2019, 2020] and theoretical [Lei et al., 2018, Mianjy et al., 2018, Rahaman et al., 2019] efforts to characterize the inductive bias implicit in these design choices, these efforts paint a partial picture or rely on impractical assumptions. If practitioners could fully characterize inductive biases implicit in learning systems, they could have a more principled way of making design decisions that increase performance and reliability [D'Amour et al., 2020].

How can we characterize the inductive biases of an arbitrary learning system in a principled, systematic manner? In this work, we do so by meta-learning Gaussian process (GP) kernel hyperparameters from the predictions of machine learning systems. Meta-learning allows us to capture shared inductive biases amongst a family of learning systems applied to a family of tasks, while GPs provide a way to make inductive biases explicit through the kernel function and the kernel hyperparameters. The kernel hyperparameters of a GP are often interpretable, yielding insights into the properties of the data on which they are trained; for example,

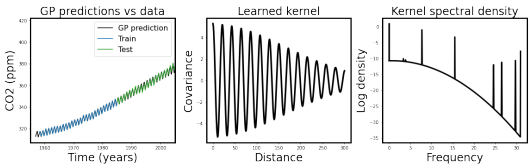


Figure 1: (**Left**) GP with spectral mixture kernel fit to the Mauna Loa CO_2 dataset. (**Middle**) The learned spectral mixture kernel. (**Right**) The spectral density (Fourier transform) of the learned kernel, revealing seasonality at different timescales.

5th Workshop on Meta-Learning at NeurIPS 2021, held virtually.

Wilson and Adams [2013] demonstrated that certain kernels can be applied to the Mauna Loa (CO_2) dataset to reveal periodic, medium-term, and long-term trends in CO_2 levels (Figure 1). If we select an expressive kernel and infer the kernel hyperparameters that are most consistent with observations of the behavior of a learning system, we can gain some insight into the inductive biases of the learning system. Furthermore, the kernel hyperparameters can serve as a compressed representation of the inductive biases of that learning system, which is useful for understanding when the system may underperform. We illustrate the potential of this framework in several case studies, ranging from the more theoretically motivated—examining the inductive biases of untrained and trained networks—to the more practically motivated—assessing similarity between architectures and assessing whether a particular neural network family is well-suited for a given dataset.

2 Background

Gaussian processes (GPs) [Rasmussen, 2003] allow us to define a distribution over functions; the GP has the property that any finite set of N observations induces a multivariate Gaussian distribution on $\mathbb{R}^N$, where the nth of these points can be interpreted as the function value, $f(\mathbf{x}_n)$, at the input point $\mathbf{x}_n$. GPs can be fully characterized by a mean function $m(\mathbf{x})$ and positive-definite kernel function $k(\mathbf{x}, \mathbf{x}')$ giving the covariance between $f(\mathbf{x})$ and $f(\mathbf{x}')$ as a function of $\mathbf{x}$ and $\mathbf{x}'$. The kernel function can be thought of as encoding an inductive bias on what kind of functions might be represented in observed data. Due to properties of the Gaussian distribution, the posterior predictive distribution at a new input, conditioned on observed data, is Gaussian with closed-form expressions for the posterior mean and variance. Model selection in Gaussian processes is typically performed through gradient-based optimization of the marginal likelihood of the data, which also admits a closed-form expression.

One choice of kernel which is expressive and differentiable in its parameters is the **spectral mixture parameterization**, introduced by Wilson and Adams [2013]. Bochner's theorem [Bochner, 1959] states that any stationary kernel and its spectral density are Fourier duals, and so every stationary kernel can be entirely characterized by a spectral density. The key insight in [Wilson and Adams, 2013] is to model the spectral density as a scale-location mixture of Gaussians. This approach has the nice theoretical property that any stationary covariance function can be approximated to arbitrary precision given sufficient mixture components and also yields a closed-form expression for the corresponding kernel, given by:

$$k(\tau) = \sum_{q=1}^{Q} w_q \cos(2\pi^2 \tau^T \mu_q)) \prod_{p=1}^{P} \exp\{-2\pi^2 \tau_p^2 v_q^{(p)}\} \,. \tag{1}$$

Here $k(\tau)$ gives the covariance between function values whose corresponding input values are a distance τ apart from each other. Here, $w = \{w_i\}_{i=1}^{Q}$ correspond to the scalar spectral mixture weights, $\mu_i \in \mathbb{R}^P$ correspond to the Gaussian mean values, $v_i \in \mathbb{R}^P$ correspond to the Gaussian variances. The mixture weight hyperparameters can be thought of as signal variances controlling the scale of the function values. The Gaussian mean values (also known as frequency parameters) can be thought of as encoding the period. The Gaussian variances can be thought of as inverse lengthscales, which capture the smoothness. Here, p iterates over the dimension of the input.

3 Inferring GP hyperparameters from neural network behavior

We aim to examine the inductive biases of a *family* $\mathfrak{F}$ of machine learning models that share in design choices (*e.g.,* the architecture, training procedure, and random initialization scheme) but differ in quantities that are randomized prior to or during training (*e.g.,* parameter initializations). In this paper, we study deep neural neural networks [LeCun et al., 2015]. We consider both untrained and trained neural networks; however, for simplicity of exposition in the remainder of this section, we describe the framework in the context of trained networks, of which untrained networks are the special case of the zeroth training iteration.

Each neural network $\mathbf{f} \in \mathfrak{F}$ is trained on a dataset $\mathcal{D}_{\text{train}} = \{\mathbf{X}^{\mathbf{t}}_{\text{train}}, \mathbf{y}^{\mathbf{t}}_{\text{train}}\}$ of samples from a function that itself belongs to a target family of functions $\mathfrak{D}$. We are interested in estimating GP kernel hyperparameters θ that best capture the *shared* inductive biases of the family $\mathfrak{F}$ of neural networks

that are trained on a target family of functions. For each model-dataset pair, the GP observes the input $\mathbf{X}_{\text{train}}$ as well as the predictions of the model $\mathbf{f}_t$ on the input, $\mathbf{f}(\mathbf{X}^{\mathbf{n}}_{\text{train}})$. Crucially, the GP does *not* observe the ground truth targets, but only the models' predictions, and thus the inferred kernel hyperparameter captures the inductive bias underlying the model family $\mathfrak{F}$ applied to the task family $\mathfrak{D}$ as evidenced by the behaviour of the model-and-task family, rather than modelling the task family directly. We note that learning GP hyperparameters is a form of meta-learning, where the estimated hyperparameters determine a shared prior over functions [Hospedales et al., 2020]. We note the approach we take is similar in motivation to Wilson et al. [2015], except in that we aim to capture the inductive biases of neural networks instead of humans.

We perform hyperparameter estimation by maximizing the log-marginal likelihoods across the model-and-task pairs:

$$\theta = \arg\max \prod_{t=1}^{T} P_\theta(\mathbf{f_t}(\mathbf{X}^{\mathbf{n}}_{\text{train}})|\mathbf{X}_{\text{train}}) = \arg\max \sum_{t=1}^{T} \log P_\theta(\mathbf{f_t}(\mathbf{X}^{\mathbf{n}}_{\text{train}})|\mathbf{X}_{\text{train}}).$$

Here $\mathbf{f}(\mathbf{X}^{\mathbf{n}}_{\text{train}})$ refers to the neural network $\mathbf{f_t}$'s predictions on $\mathbf{X}^{\mathbf{n}}_{\text{train}}$. Each term in the sum is the marginal likelihood of a neural network's predictions. Choosing this objective corresponds to an assumption that neural networks from a particular family have meaningful shared inductive biases. One way to justify this is to consider randomly initialized convolutional neural networks: Although the networks themselves are different insofar as they are computing different functions, they all exhibit the property of translational invariance. Furthermore, this perspective is in fact precise in the infinite width limit as infinite-width randomly initialized neural networks (with particular architectures) are Gaussian processes whose kernels (which capture the inductive biases) admit analytic expressions or can be computed numerically [Williams, 1996, Neal, 1996, Lee et al., 2017].

A technique we will make use of repeatedly is inspecting the learned kernel hyperparameters to extract insights into data. We briefly remark on the main advantages of analyzing the kernel as opposed to inspecting the data itself. The kernel hyperparameters represent a precise, quantitative summary of the inductive biases and can capture properties that are difficult to see from the learned function itself. Furthermore, as we will demonstrate later, the kernel hyperparameters are themselves a representation that can be utilized in a variety of ways.

4 Experiments

Experiments 1 and 2 establish that our framework can reliably capture known inductive biases. Additional experiments (Experiments 3 and 4) demonstrate the applications of our approach to precisely characterizing inductive biases and then using using these precise characterizations to make predictions about which models are well-suited for which datasets.

4.1 Reproduction: Capturing spectral bias in neural networks

We first assess whether the spectral mixture kernel accurately captures inductive biases of neural networks in a situation where the inductive biases are well-known. In particular, we study a phenomenon discussed in Rahaman et al. [2019], who demonstrated that neural networks tend to learn low-frequency signals in the target function before high-frequency signals. To illustrate this phenomenon, the authors trained a 6-layer, 256-width neural network with ReLU activations on a one-dimensional function consisting of a sum of sinusoidal functions with varying frequencies. By examining the Fourier spectrum of the network predictions, the authors show that lower frequencies are indeed learned earlier.

In Figure 2, we plot the predictions of the same neural network from Rahaman et al. [2019] as training progresses. We also plot the covariance (kernel value) as a function of distance between two points (where the kernel is the spectral mixture kernel with hyperparameters fit to the neural network predictions). Intuitively, this plot shows us how the similarity between function values varies with the distance between their input points.[1] Initially the learned kernel only shows evidence of a low-frequency signal in the trained networks predictions. However, as training progresses, the learned kernel has picked up on both low-frequency and high-frequency signals. The changes in

[1]Note, since the spectral mixture kernel is stationary, only the distance between points matters.

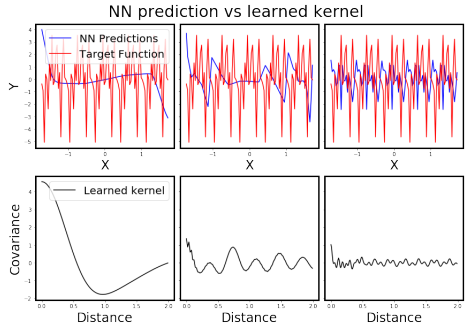


Figure 2: (**Top row**) Neural network predictions as training progresses on a target function consisting of a sum of sines with different frequencies. (**Bottom row**) Spectral mixture kernel fit to a subset of neural network predictions as training progresses.

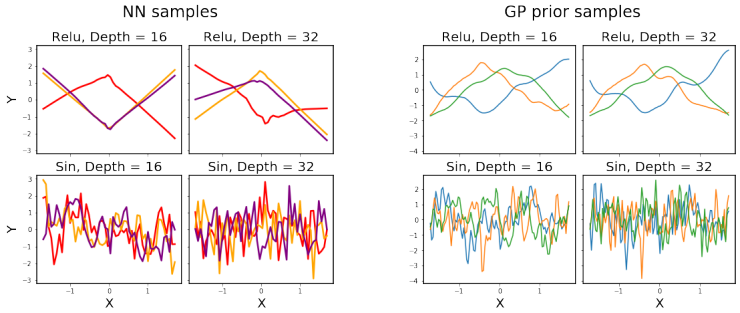


Figure 3: (**Left**) Functions represented by NNs. (**Right**) Samples from a GP prior with kernel hyperparameters inferred from the NN predictions in left panel.

the learned kernel as training progresses are consistent with the findings reported by Rahaman et al. [2019].[2]

4.2 Reproduction: Validating consistency of GP priors with NN priors

In this section, we verify that the learned kernels (inferred from neural networks) produce a GP prior that is qualitatively consistent with the corresponding inductive bias of the neural network. One simple way to verify this is to compare neural networks alongside samples from a GP prior with kernel hyperparameters inferred from those neural network predictions. In Figure 3, we plot a few samples from the GP prior with hyperparameters inferred from randomly initialized neural network predictions across a small range of widths and depths and across the Sin and ReLU activations. Unsurprisingly, the spectral mixture kernel accurately captures the periodicity of the Sin activation networks. In contrast, the spectral mixture kernel cannot produce piecewise linear functions. However, the sampled functions do reproduce some of the qualitative properties observed in the ReLU network predictions such as the cusp and two distinct sections of the input domain where the ReLU networks are either monotonically increasing or decreasing. This example also highlights that the spectral mixture parameterization is highly expressive, able to capture a wide range of inductive biases.

4.3 Investigating priors in randomly-initialized neural networks

The previous two sections illustrate that the spectral mixture kernel is a viable tool for interrogating inductive biases of neural networks. In this section, we apply the spectral mixture kernel to analyzing settings in which the inductive biases are less well-understood. In particular, we empirically study

[2]We find that the marginal likelihood for the spectral mixture kernel is highly multimodal in its frequency parameters. For this particular problem, the results are sensitive to the random initialization. In future work, we will apply well-established techniques like (approximate) marginalization of the hyperparameters that are known to alleviate this issue [Simpson et al., 2020].

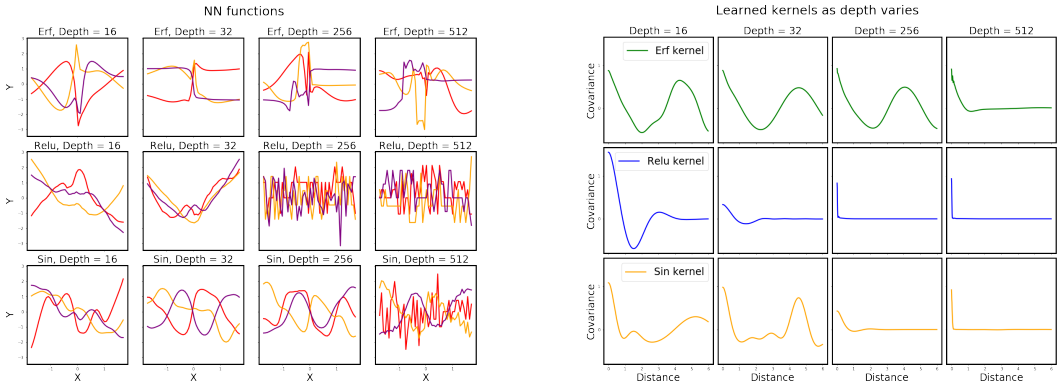


Figure 4: (**Left four columns**) Samples from neural networks across different activation and depths. (**Right four columns**) Learned kernels across different activations and depths.

the behavior of randomly-initialized, finite-width, finite-depth networks as we vary depth. This setting is of interest because deeper networks perform better empirically, but the kinds of changes in the inductive bias that lead to this better performance are not yet well-understood.

In the first set of experiments, we sample randomly initialized networks across different activations and depths and then fit spectral mixture kernels to the neural network predictions. [3] The number of hidden units in each layer is 128. We study three sets of activations: ReLU, Erf, and Sin.[4]

Figure 4 plots the learned kernels for each activation across different depths as well as samples from the corresponding neural network families. For the Sin activation, we see that as we increase depth from 16 to 32, the kernel picks up on long-range correlations. Across all activations, the learned kernels reveal an interesting pathology: for large depths, the learned kernel sharply decays towards zero as distance between points increases. This is consistent with what we see in the sampled functions on the right of Figure 4. The deep networks become quickly-varying everywhere in the input domain which is why the spectral mixture kernel has learned very short lengthscales. This pathology is consistent with what has been reported in several papers [Duvenaud et al., 2014, Schoenholz et al., 2017].

4.4 Predictability: Using kernel hyperparameters as representation of inductive biases

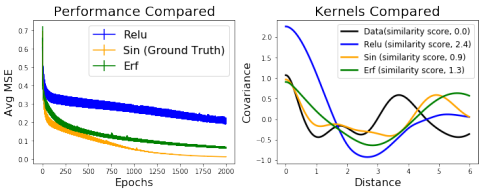
In this section, we assess whether the framework introduced in this paper can address the following question: if the inductive biases of a network and characteristics of a dataset are well-matched, does the model perform better? We are well-equipped to investigate this question because we have a framework that allows us to extract a representation, in the form of the inferred kernel hyperparameters, of the inductive biases of a model as applied to a given dataset. In particular, for a given family of target functions, we fit a set of kernel hyperparameters directly to the ground-truth observations, termed the *data hyperparameters*. We also, as we have done in previous sections, infer kernel hyperparameters on the basis of the behavior of a particular family of neural networks, which we call the *model hyperparameters*. A natural question to ask is: when the model hyperparameters and data hyperparameters (both kernels in our context) are well-matched, do we see better test set performance?

We investigate this question in two experiments with two different families of target functions. The first family of target functions consists of samples from a GP with a spectral mixture kernel which were chosen to match the inductive biases of Sin activation networks. The second family of target functions consists of randomly initialized ReLU network.[5]. These target functions were chosen primarily because the properties of these functions can be precisely understood and serve as useful sanity checks. The networks we compare in this analysis have width of 16, depth of 4, and weight

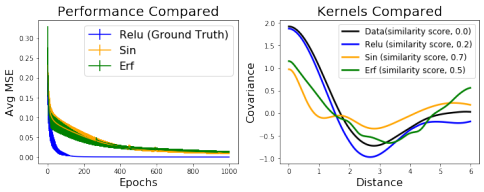
[3] For the weight initialization, we use the NTK parameterization (with weight variance $\sigma_w^2 = 1.0$, bias variance $\sigma_b^2 = 0.05$) of Jacot et al. [2018], Novak et al. [2020].

[4] Sin: $a\sin(bx + c)$, ReLU: $max(0, x)$, Erf: $a\,\mathrm{erf}(bx) + c$ where $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,\mathrm{d}t$.

[5] The networks have a width of 16, depth of 4, and weight and bias variances of 1.00 and 0.05.



(a) (Left) Test set performance averaged across three families of networks; the target function is samples from a spectral mixture kernel with strong periodicity. **(Right)** The learned kernels for each family alongside the kernel inferred from the target function.



(b) (Left) Test set performance averaged across three families of networks; the target functions are ReLU networks. **(Right)** The learned kernels for each family alongside the kernel inferred from the target function.

and bias variances of 1.00 and 0.05 respectively. We compute similarity between two kernels k and k' by comparing their Gram matrices using the relative Frobenius norm of Stephenson et al. [2021], $\|k(X,X) - k'(X,X)\|_F \,/\, \|k'(X,X))\|_F$. Figure 5a plots the test-set MSE averaged across the family of networks we train as well as the kernels corresponding to the data and model hyperparameters. The ranking of the networks by test set performance (Sin, Erf, ReLU) is consistent with the similarity scores (as defined by the Frobenius norm) between the model kernels and the data kernel.

Figure 5b repeats the same analysis except on the ReLU target functions. The similarity between the ReLU model kernel and the data kernel (which also consists of ReLU networks) is the highest. Indeed, we do see that ReLU networks achieve the lowest test error. However, the kernel similarity scores predict that Erf will achieve a lower test error than Sin; in reality, there is not a significant difference between the Sin and Erf final test losses. One explanation for the inconsistency is the mismatch in inductive biases between the spectral mixture kernel and the Erf networks. The Erf network is non-stationary, as it is smooth and slowly-varying for most of its input domain but has a sharp kink. In order to capture this behavior, the spectral mixture kernel learns a short lengthscale; since the spectral mixture kernel is stationary it cannot learn a function whose smoothness varies with the input domain. As a result, the inferred kernel hyperparameters are not consistent with the inductive biases of the Erf networks.[6]

5 Discussion

In this paper, we illustrated the potential of meta-learning Gaussian process hyperparameters as a means of quantifying inductive biases in a learning system. In a diverse range of settings, we were able to accurately capture inductive biases in neural network models via GP kernel hyperparameters, and utilize the hyperparameters to assess whether a model family is compatible with a particular task family.

There are many natural extensions of this work, including scaling the methodology to more complex tasks, but we will focus on one in this discussion. In this work, we have adopted the perspective of thinking of kernel hyperparameters as a representation of neural network inductive biases. Given this perspective, one natural question is how do the inferred kernel hyperparameters change if we apply certain transformations to the inputs and outputs of the dataset we learn the hyperparameters from? This line of questioning is inspired by work such as [Kornblith et al., 2019].

While the inferred kernel hyperparameters depend on a complicated optimization procedure about which it is difficult to make precise mathematical statements, we can make a few remarks about some of the invariances that the GP marginal likelihood (the objective function we use to learn the kernel hyperparameters) exhibits. One is invariance to orthogonal transformations of the input vectors. We begin by noting that the Gram matrix K of a sequence of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is invariant under orthogonal transformation of those vectors because inner products are preserved under orthogonal transformation. Since the input vectors appear in the GP marginal likelihood only through the Gram matrix, the marginal likelihood is invariant under orthogonal transformations of the input vectors, and thus the kernel hyperparameter representation is invariant as well. Future work will identify whether this property is desirable.

[6] See [Rasmussen and Williams, 2006] for a similar illustration of mismatch in inductive biases between kernel and data with the radial basis function kernel fit to a step function.

6 Appendix

References

Salomon Bochner. *Lectures on Fourier integrals*. Princeton University Press, 1959.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

David Duvenaud, Oren Rippel, Ryan P. Adams, and Zoubin Ghahramani. Avoiding pathologies in very deep networks. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2014. URL `http://www.cs.princeton.edu/~rpa/pubs/duvenaud2014pathologies.pdf`. arXiv:1402.5836 [stat.ML].

Timothy M Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8580–8589, 2018. URL `https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html`.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited, 2019.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. In *nternational Conference on Learning Representations*, 10 2017.

Deren Lei, Zichen Sun, Yijun Xiao, and William Yang Wang. Implicit regularization of stochastic gradient descent in natural language processing: Observations and implications. *ArXiv*, abs/1811.00659, 2018.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *ACL*, 2019.

R Thomas McCoy, Robert Frank, and Tal Linzen. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140, 2020.

Poorya Mianjy, Raman Arora, and Rene Vidal. On the implicit bias of dropout. In *International Conference on Machine Learning*, pages 3540–3548. PMLR, 2018.

Radford M. Neal. *Priors for Infinite Networks*, pages 29–53. Springer New York, New York, NY, 1996. ISBN 978-1-4612-0745-0. doi: 10.1007/978-1-4612-0745-0_2. URL `https://doi.org/10.1007/978-1-4612-0745-0_2`.

Roman Novak, Lechao Xiao, Jiri Hron, Jaehoon Lee, Alexander A. Alemi, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in python. In *International Conference on Learning Representations*, 2020. URL `https://github.com/google/neural-tangents`.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/rahaman19a.html`.

Carl E. Rasmussen and Christopher K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, January 2006.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation, 2017.

F. Simpson, V. Lalchand, and C. Rasmussen. Marginalised gaussian processes with nested sampling. *ArXiv*, abs/2010.16344, 2020.

William T. Stephenson, Soumya Ghosh, Tin D. Nguyen, Mikhail Yurochkin, Sameer K. Deshpande, and Tamara Broderick. Measuring the sensitivity of gaussian processes to kernel choice, 2021.

Christopher K. I. Williams. Computing with infinite networks. In *NeurIPS*, 1996.

Andrew G Wilson, Christoph Dann, Chris Lucas, and Eric P Xing. The human kernel. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper/2015/file/4462bf0ddbe0d0da40e1e828ebebeb11-Paper.pdf`.

Andrew Gordon Wilson and Ryan P. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013. URL `http://www.cs.princeton.edu/~rpa/pubs/wilson2013kernels.pdf`. arXiv:1302.4245 [stat.ML].