DEEP KERNELS WITH PROBABILISTIC EMBEDDINGS FOR SMALL-DATA LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

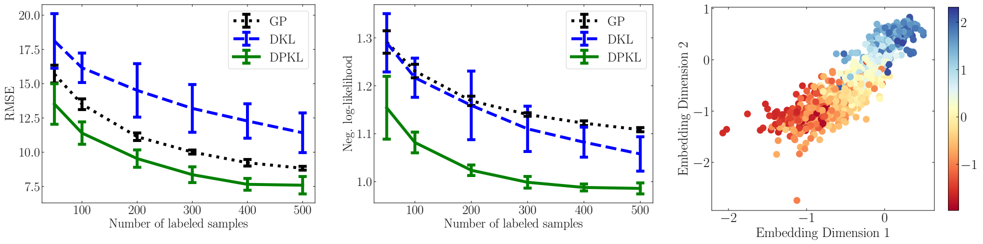
Gaussian Processes (GPs) are known to provide accurate predictions and uncertainty estimates in small-data settings by capturing similarity between data points through their kernel function. However, traditional GP kernels do not work well with high dimensional data points. A solution is to use a neural network to map data to low dimensional embeddings before kernel computation. However, the huge data requirement of neural networks makes this approach ineffective in small-data settings. We solve the conflicting issues of representation learning and data efficiency, by mapping high dimensional data to low dimensional probability distributions using a probabilistic neural network and then computing kernels between these distributions to capture similarity. We also derive a functional gradient descent approach to enable end-to-end training of our model. Experiments on various datasets show that our approach outperforms the state-of-the-art in GP kernel learning.

1 INTRODUCTION

Huge labeled datasets (Deng et al., 2009) have played a key role in the recent success of deep learning (Krizhevsky et al., 2012; Graves et al., 2013). However, in many emerging applications of machine learning like materials science (Zhang et al., 2020) and poverty prediction (Jean et al., 2016) large labeled training datasets may not be available leading to a drop in model accuracy. Additionally, deep neural networks generally do not provide accurate uncertainty estimates, leading to over-confident but incorrect predictions (Bulusu et al., 2020) in small-data settings.

Gaussian Processes (GPs) (Rasmussen, 2003) are non-parametric models that leverage correlation (similarity) between data points to give a probabilistic estimate of the target values. Exploiting similarity in a non-parametric fashion contributes to sample efficiency, while probabilistic estimation helps to quantify model uncertainty in unlabeled regions of the input space. However, owing to the curse of dimensionality, the number of points required to model the covariance using traditional GP kernels grows exponentially with data dimension (Tripathy et al., 2016) which reduces their efficacy in the *small-data* regime for high-dimensional data. The success of deep neural networks in learning low dimensional representations of high dimensional data has led to works (Wilson et al., 2016) that use a neural network to map data to a low dimensional latent space and then build a GP for prediction on the *latent space*. However, owing to the huge data requirement of neural networks, these approaches are also not as effective in learning latent representations in the *small-data regime*.

To enable data-efficient representation learning, we propose a new approach for learning highly expressive GP kernels with small amounts of labeled data by mapping data points to *probability distributions* in a latent space with a probabilistic neural network (Neal, 2012). This is motivated by prior works showing that probabilistic models outperform deterministic models when the amount of data is insufficient to capture the complexity of the task (Wilson & Izmailov, 2020; Gal, 2016). We then use the theory of kernel embeddings of distributions (Muandet et al., 2017) to build expressive kernels, and consequently GPs, on the latent probability distributions. To enable end-to-end learning, we derive a functional gradient descent procedure for training the model via maximum likelihood estimation of the *distribution over model parameters*. Results in Figure 1 show that our model, Deep Probabilistic Kernel Learning (DPKL), outperforms baselines – Deep Kernel Learning (DKL) (Wilson & Nickisch, 2015), and a GP with a Squared Exponential (SE) kernel, in terms of both prediction error [lower Root Mean Squared Error (RMSE) in Figure 1a], and uncertainty quantification [lower negative log-likelihood in Figure 1b] while learning meaningful embeddings [Figure 1c].



(a) Accurate Prediction (b) Uncertainty Quantification (c) Representation Learning

Figure 1: Performance of our models and baselines for regression on the UCI (Lichman et al., 2013) CTSlice Dataset (Dimensionality = 384) with $n = \{50, 100, 200, 300, 400, 500\}$ labeled data points. (a) Our approach DPKL, has *significantly lower* RMSE than conventional GPs and Deep Kernel Learning (DKL). (b) DPKL also captures the data distribution much better (lower negative log likelihood). (c) Points with similar target values have similar mean embeddings in the 2-D latent space learned by DPKL with $n = 100$ labeled samples.

2 Background and Related Work

Given n training points, $\mathbf{X} \in \mathcal{X} \subseteq \mathbb{R}^{n \times D}$, and targets, $\mathbf{y} \in \mathbb{R}^{n \times 1}$, our goal is to accurately predict targets y_* for test data points $\mathbf{x}_*$ especially when there is limited training data (small n) and the dimensionality (D) is large.

Gaussian Processes. A Gaussian Process (GP) (Rasmussen, 2003) defines a *probability distribution* over functions $H : \mathbb{R}^d \to \mathbb{R}$ such that individual function values form a multivariate Gaussian distribution i.e. $H(\mathbf{X}) = [H(\mathbf{x}_1), \ldots, H(\mathbf{x}_n)] \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$ with entries of the mean vector given by $\mu_i = \mu(\mathbf{x}_i)$, and of the covariance matrix given by $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. A GP is fully specified by the mean function μ (typically $\mu \equiv 0$) and the covariance kernel K (For eg. SE or Matern) which measures similarity between data points. Data is also assumed to be corrupted by noise $\eta \sim \mathcal{N}(0, \sigma_\eta^2)$ and the overall covariance is – $\mathrm{Cov}(H(\mathbf{x}_i), H(\mathbf{x}_j)) = K(\mathbf{x}_i, \mathbf{x}_j) + \sigma_\eta^2 \mathbb{1}\{i = j\}$. Given the training data $\mathbf{X}$ and associated targets $\mathbf{y}$, the predicted target value $H(\mathbf{x}_*)$ at a test point $\mathbf{x}_*$ follows a Gaussian distribution, i.e. $\Pr(H(\mathbf{x}_*)|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) = \mathcal{N}(\mu(\mathbf{x}_*), \sigma^2(\mathbf{x}_*))$ where $\mu(\mathbf{x}_*) = \mathbf{k}_*{}^T(\mathbf{K} + \sigma_\eta^2)^{-1}\mathbf{y}$ and $\sigma^2(\mathbf{x}_*) = k_{**} - \mathbf{k}_*{}^T(\mathbf{K} + \sigma_\eta^2)^{-1}\mathbf{k}_*$ with $\mathbf{k}_* = K(\mathbf{X}, \mathbf{x}_*)$ and $k_{**} = K(\mathbf{x}_*, \mathbf{x}_*)$.

Deep Kernel Learning. Deep Kernel Learning (DKL) (Wilson et al., 2016) uses a neural network to learn a GP kernel as $\mathbf{K}_{ij} = K(g_\mathbf{w}(\mathbf{x}_i), g_\mathbf{w}(\mathbf{x}_j))$ where K is a standard GP kernel and $g_\mathbf{w}(.)$ represents the neural network with parameters $\mathbf{w}$ which can be learned by minimizing the GP negative log likelihood. While DKL provides greater flexibility than standard GP kernels, it inherits the limitations of neural networks and requires a large amount of labeled data for accurate prediction.

Probabilistic Neural Networks. Probabilistic models learn a distribution over model parameters thus incorporating uncertainty in predictive modeling which can improve generalization performance particularly when there is insufficient data (Wilson & Izmailov, 2020). Unlike prior works (Graves, 2011) that train probabilistic or Bayesian Neural Networks (BNNs) for predictions, we use them to improve the quality of low dimensional embeddings of high dimensional data. The probabilistic embeddings given by our model serve as inputs to predictive models like GPs. Thus our approach aims to improve GP kernel learning and not replace BNNs.

Other related work. Deep GPs (Damianou & Lawrence, 2013) seeks to improve the performance of GPs in the *big-data* regime by hierarchically stacking multiple GPs. They are known to perform worse than single layer GP for small datasets (Salimbeni & Deisenroth, 2017) while we show that our model DPKL improves upon single layer GP in these settings. GPs on distribution inputs have also been theoretically studied in (Bachoc et al., 2017; 2018). They assume that the input itself is a distribution while we use neural networks to learn distributional embeddings of deterministic inputs.

3 Deep Probabilistic Kernel Learning

We briefly describe our model – Deep Probabilistic Kernel Learning (DPKL) (further details in Appendix A). DPKL predicts output $y \in \mathbb{R}$ given input $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^D$ using a Gaussian Process (GP) learned over *probability distributions* in a low dimensional latent space $\mathcal{Z} \subseteq \mathbb{R}^d$ ($d << D$) where distributions in $\mathcal{Z}$ are obtained by passing the input data through a probabilistic neural network.

Probabilistic Latent Space Mapping. Each data point $\mathbf{x}$ passes through a probabilistic neural network with parameters $\mathbf{W} \sim p(\mathbf{W})$ to give a random variable $\mathbf{Z} = g_{\mathbf{W}}(\mathbf{x}) \in \mathcal{Z}$. Thus a point $\mathbf{x}_i$ is represented by the latent distribution $p(\mathbf{Z}|\mathbf{x}_i)$ due to the stochasticity of $\mathbf{W}$.

GP Regression over Latent Distributions. We assume a GP prior $H \sim \mathcal{GP}(0, \mathbf{K})$ over target functions H which take *distributions* $p(\mathbf{Z}|\mathbf{x}_i)$ as input and output $\hat{y}_i$, an estimate of the true label y_i. $\mathbf{K}$ is the kernel matrix which models covariance between GP inputs (probability distributions). Given any two probability distributions $p(\mathbf{Z}|\mathbf{x}_i)$ and $p(\mathbf{Z}|\mathbf{x}_j)$, the corresponding entry of $\mathbf{K}$ is given by $K_{ij} = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{Z}|\mathbf{x}_i),\, \mathbf{z}'\sim p(\mathbf{Z}|\mathbf{x}_j)}[k(\mathbf{z}, \mathbf{z}')]$. Here k can be any standard kernel like the SE kernel. This choice of kernel between probability distributions follows previous works (Muandet et al., 2012; 2017) which use similar kernels to build predictive models, such as, SVMs, with distribution inputs. We expect that *probability distributions* in the latent space $\mathcal{Z}$ will better capture uncertainty due to scarcity of training data than point embeddings. Hence, we perform GP regression over distributions in $\mathcal{Z}$.

Training Algorithm. The forward pass described above is used to compute the kernel between data points and consequently the predicted GP mean and variance for test data (see section 2). To train the model we need to find the optimal distribution $p(\mathbf{W})$ over its parameters in this setting.

Since the latent embedding $\mathbf{Z}$ for a data point $\mathbf{x}$ is given by $\mathbf{Z} = g_{\mathbf{W}}(\mathbf{x})$, $\mathbf{W} \sim p(\mathbf{W})$, we can express the entries of the kernel matrix as

$$K_{ij} = \mathbb{E}_{\mathbf{w},\mathbf{w}'\sim p(\mathbf{W})}[k(g_{\mathbf{w}}(\mathbf{x}_i), g_{\mathbf{w}'}(\mathbf{x}_j))] = K_{ij}[p] \tag{1}$$

We can view K_{ij} as a *functional* of $p(\mathbf{W})$. The overall data likelihood is then also a functional of p and the negative log likelihood is given by (see (Rasmussen, 2003))

$$L[p] = -\log \Pr(\mathbf{y}|\mathbf{X}, p(\mathbf{W})) = \frac{1}{2}\mathbf{y}^T(\mathbf{K}[p] + \sigma^2\mathbf{I})^{-1}\mathbf{y} + \frac{1}{2}\log(\det(\mathbf{K}[p] + \sigma^2\mathbf{I})) \tag{2}$$

where $\det(\mathbf{A})$ denotes the determinant of a matrix $\mathbf{A}$.

Thus a *maximum likelihood* estimate, $p^*(\mathbf{W})$, of the distribution over model parameters is given by

$$p^*(\mathbf{W}) = \underset{p(\mathbf{W})\in\mathcal{P}}{\arg\min}\, L[p] \tag{3}$$

where $\mathcal{P}$ is the class of distributions in which we seek p^*.

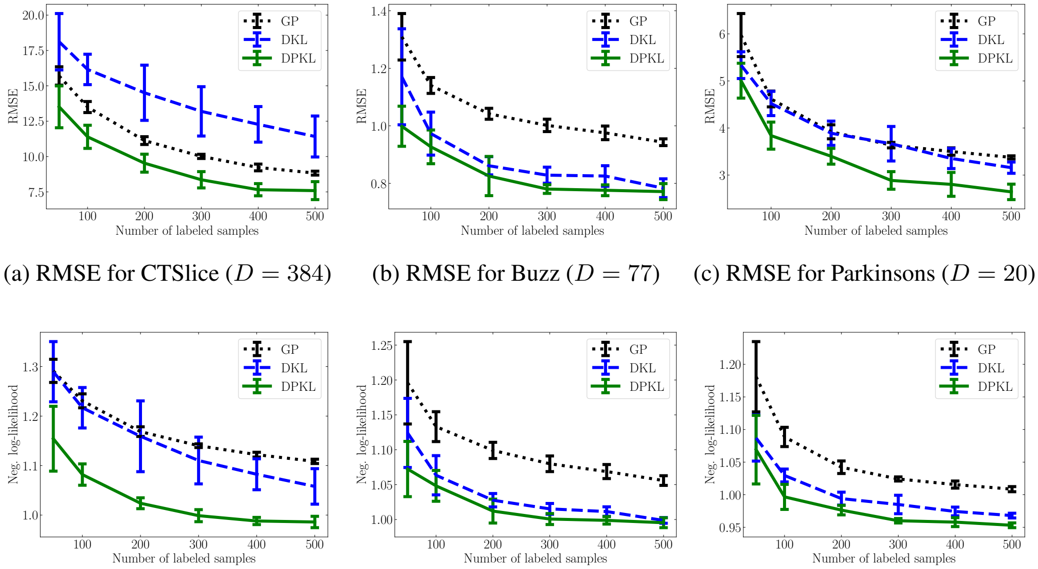
The choice of $\mathcal{P}$ is critical to the success of this approach. Following Liu & Wang (2016), we choose $\mathcal{P}$ to be $\mathcal{P} = \{p(\mathbf{u})|\mathbf{u} = \mathbf{w} + s(\mathbf{w}), \mathbf{w} \sim p_0(\mathbf{w}), s \in \mathcal{H}_\kappa\}$ where $\mathcal{H}_\kappa$ is a RKHS given by a kernel κ between model parameters $\mathbf{w}$ (note that this is *different* from the kernel k between latent embeddings in Equation (1)). This choice of $\mathcal{P}$ includes all smooth transformations $\mathbf{u} = \mathbf{w} + s(\mathbf{w})$ of samples $\mathbf{w}$ drawn from an initial distribution $p_0(\mathbf{w})$, and the optimization problem in (3) now reduces to computing the optimal shift $s^*(\mathbf{w})$. Next, we provide an expression for the functional gradient of the negative log-likelihood under our model with respect to the shift s.

Proposition 1. *If we draw m realizations of model parameters $\mathbf{w}_1, \ldots, \mathbf{w}_m \sim p(\mathbf{w})$, $p \in \mathcal{P}$, then*

$$\nabla_s L\mid_{s=0} \simeq \sum_{l=1}^{m} \kappa(\mathbf{w}_l, .)\nabla_{\mathbf{w}_l}\hat{L}(\mathbf{w}_1, \ldots, \mathbf{w}_m) \tag{4}$$

where $\hat{L}$ is given by substituting $\hat{K}_{ij} = \frac{1}{m^2}\sum_{l,l'} k(g_{\mathbf{w}_l}(\mathbf{x}_i), g_{\mathbf{w}_{l'}}(\mathbf{x}_j))$ in Equation (2).

To estimate the optimal shift, s^* (or optimal distribution p^*) we draw an initial set of parameters $\mathbf{w}_1, \ldots, \mathbf{w}_m \sim p_0(\mathbf{w})$ and iteratively apply the functional gradient descent transformation $\mathbf{u} = \mathbf{w} - \epsilon\nabla_s L\mid_{s=0}$ as described in Algorithm 1 in Appendix A.



(a) RMSE for CTSlice ($D = 384$) (b) RMSE for Buzz ($D = 77$) (c) RMSE for Parkinsons ($D = 20$)

(d) Negll for CTSlice ($D = 384$) (e) Negll for Buzz ($D = 77$) (f) Negll for Parkinsons ($D = 20$)

Figure 2: Results for regression on 3 UCI Datasets with $n = \{50, 100, 200, 300, 400, 500\}$ labeled samples. Plots (a) - (c) show that DPKL has lower RMSE than baselines - DKL(Wilson et al., 2016) and GP. Plots (d) - (f) show that DPKL quantifies uncertainty better (lower average negative log-likelihood (Negll) of test data). See Appendix B for results on more datasets.

4 Experimental Results

We apply our model, DPKL, to regression tasks in the small data regime on 6 datasets of varying dimensionality from the UCI repository (Lichman et al., 2013). We compare DPKL to DKL (Wilson et al., 2016) (which uses a deterministic neural network to map data into the latent space), and a simple GP which uses a Squared Exponential (SE) kernel directly on the data. The lengthscales for the GP SE kernel, and the neural network weights in DKL are optimized using gradient descent on the negative log-likelihood while the *distribution* over neural network weights in DPKL is optimized using functional gradient descent as described in Section 3. For each dataset we use $n = \{50, 100, 200, 300, 400, 500\}$ labeled examples. Additional details are in Appendix B.

Figure 2 shows results for 3 datasets. Clearly DPKL has lower prediction error (lower RMSE) than DKL and GP for all datasets. The improvements in the high dimensional datasets CTSlice ($D = 384$) and Buzz ($D = 77$), clearly illustrate the advantages of our approach in high dimensional settings. Moreover, our approach also provides better uncertainty quantification measured by the average negative log likelihood of test data (Lakshminarayanan et al., 2017). A lower negative log likelihood implies that the model fits the test data better. Results in Figure 2 shows that by projecting data to latent probability distributions DPKL consistently outperforms DKL and GP in this metric.

5 Conclusion

We proposed a new approach for small-data learning that maps high dimensional data to low dimensional probability distributions and then performs GP regression on these distributions. The distribution over model parameters is learned via functional gradient descent. Our model outperforms several baselines in GP regression while learning a meaningful representation of the data and accurately quantifying uncertainties on test data. In future, we plan to theoretically analyze the convergence of our approach, seek potential improvements through optimal kernel selection and unbiased gradient estimation, and apply our model to areas like Bayesian Optimization (Brochu et al., 2010) where GPs have been successful in the past.

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

François Bachoc, Fabrice Gamboa, Jean-Michel Loubes, and Nil Venet. A gaussian process regression model for distribution inputs. *IEEE Transactions on Information Theory*, 2017.

Francois Bachoc, Alexandra Suvorikova, David Ginsbourger, Jean-Michel Loubes, and Vladimir Spokoiny. Gaussian processes with multidimensional distribution inputs via optimal transport and hilbertian embedding. *arXiv preprint arXiv:1805.00753*, 2018.

Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K Varshney, and Dawn Song. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020.

Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pp. 207–215, 2013.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Yarin Gal. Uncertainty in deep learning. *University of Cambridge*, 2016.

Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pp. 2348–2356, 2011.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649. IEEE, 2013.

Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

Neal Jean, Sang Michael Xie, and Stefano Ermon. Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. In *Advances in Neural Information Processing Systems*, pp. 5322–5333, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Moshe Lichman et al. Uci machine learning repository, 2013.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pp. 2378–2386, 2016.

Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. Learning from distributions via support measure machines. In *Advances in neural information processing systems*, pp. 10–18, 2012.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pp. 63–71. Springer, 2003.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 4588–4599, 2017.

Rohit Tripathy, Ilias Bilionis, and Marcial Gonzalez. Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation. *Journal of Computational Physics*, 321:191–223, 2016.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pp. 1775–1784, 2015.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pp. 370–378, 2016.

Jize Zhang, Bhavya Kailkhura, and T Han. Leveraging uncertainty from deep learning for trustworthy materials discovery workflows. *arXiv preprint arXiv:2012.01478*, 2020.