Deep Learning and Gaussian Processes: Some Connections

Ansu Chatterjee

School of Statistics, University of Minnesota

MAR 2, 2021

Caveat:

This talk is (mostly) an overview!

Why bother?

- (**Proven skill:**) Deep neural networks (DNNs) have shown fantastic performance in *discriminative* tasks, in large datasets, as a black box.

Why bother?

- (**Proven skill:**) Deep neural networks (DNNs) have shown fantastic performance in *discriminative* tasks, in large datasets, as a black box.

- (**Interpretable DNN:**) In order to be useful for science, we need to understand what is DNN doing.

Why bother?

- (**Proven skill:**) Deep neural networks (DNNs) have shown fantastic performance in *discriminative* tasks, in large datasets, as a black box.
- (**Interpretable DNN:**) In order to be useful for science, we need to understand what is DNN doing.
- (**Scientific method:**) In particular, we need to understand what features it is selecting or up/down weighing, and how to test hypothesis, how to make DNN inferences reproducible.
- (**Need probability:**) *We need a probabilistic framework coupled with DNNs.*

Proven skills of DNN

- (**Universal approximation:**) (Informally) under reasonable assumptions, a 2-layer network with a hidden layer of *arbitrary width* can approximate a continuous function with arbitrary precision.

- Several variations of the above are established for finite width, *arbitrary depth* cases. (One was officially published in January, 2021!)

A typical result for deep belief networks

Theorem (Hinton *et al.*, Le Roux-Bengio, Montufar-Ay, . . .)

(Informally,) a deep belief network of depth $\approx 2^{n-1}$ *and width* n *can approximate any probability distribution on* $\{0,1\}^n$ *arbitrarily closely.*

A deep but narrow network does not require more parameters compared to a shallow, wide network.

What does this tell us?

- A universal approximation result is a minimal requirement. A shallow network of *arbitrary width* is the baseline standard.

- This does not tell us anything about interpretability of the network, has no information on the rate of approximation and UQ, this is available only for a limited number of deep belief and other networks.

- *What else do we know about shallow, wide networks?*

The GP connection

Theorem (Neal (1992), Williams (1997), . . .)

(Informally,) a fully connected artificial neural network with one hidden layer of arbitrary width, and independent random weights and biases acts like a Gaussian process (GP).

The proof uses the δ-method (from Statistics) and the Central Limit Theorem.

What is a GP?

A stochastic process $X : T \to \mathbb{R}$ is a Gaussian Process (GP) if there exists a *mean function* $\mu : T \to \mathbb{R}$ and a ***a positive definite kernel function*** $K : T \times T \to \mathbb{R}$, such that
for ***any positive integer*** k, and
for ***any selection of*** $\{t_1, t_2, \ldots, t_k\} \subset T$,

$$\begin{pmatrix} X(t_1) \\ X(t_2) \\ . \\ . \\ . \\ X(t_k) \end{pmatrix} \sim N_k \left(\begin{pmatrix} \mu(t_1) \\ \mu(t_2) \\ . \\ . \\ . \\ \mu(t_k) \end{pmatrix}, \begin{pmatrix} K(t_1,t_1) & K(t_1,t_2) & \ldots & K(t_1,t_k) \\ K(t_2,t_1) & K(t_2,t_2) & \ldots & K(t_2,t_k) \\ . & . & \ldots & . \\ . & . & \ldots & . \\ . & . & \ldots & . \\ K(t_k,t_1) & K(t_k,t_2) & \ldots & K(t_k,t_k) \end{pmatrix} \right).$$

The DL-GP trade-off

- DNNs achieve excellent *discriminative performance* on many real-world problems.
- DNNs typically are *scalable*.
- DNNs may overfit on small datasets, may not be interpretable and do not have UQ.

The DL-GP trade-off

- GPs are suitable for small datasets and are *suitable for statistical inference* including hypothesis testing, confidence and prediction bounds, UQ.
- GPs are typically not scalable and choosing a good kernel in practice is challenging.
- A DL-GP connection can combine strengths and bring the best of both worlds!

The numerous DL-GP connections!

DL-GP

There are **many** ways of linking DNNs and GP!

- Using infinite width asymptotics, as done classically.
- Use infinitely wide as well as deep networks.

The numerous DL-GP connections!

DL-GP

There are **many** ways of linking DNNs and GP!

- Using narrow but deep fully connected networks and Gaussian weights and biases.
- By using randomized activation functions.
- By leveraging the cost function.

The numerous DL-GP connections!

DL-GP

There are **many** ways of linking DNNs and GP!

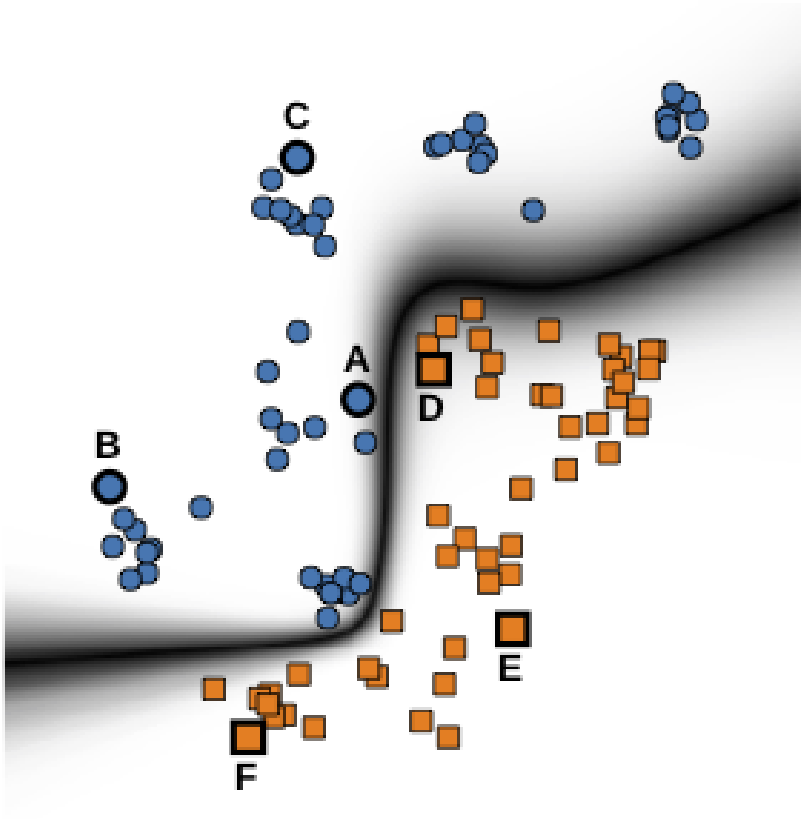
- (Deep GPs:) By stacking stochastic processes inside DNNs.
- (Dropouts:) a very popular scheme that ties DNNs with GPs.
- (Deep kernels for GPs:) build kernels using DNNs for use inside GPs.

DL-GP connection: the neural tangent kernel

DL-GP

There are **many ways of linking DNNs and GP!**

A **neural tangent kernel** (NTK) arises in the context of a kernel gradient descent approach for training, and as the width of the DNN tends to infinity, this asymptotes to a GP. Other training and optimization schemes similarly tie-in a GP to a DNN.



DL-GP connection: the NNGP

- Construct a fully-connected MLP with independent random weights and biases.
- A recursive formula allows us to compute the covariances between the neurons at the output layer. As all layer widths increase to infinity, this converges to a GP kernel.
- Training and inference is done by using this (approximate) GP structure, instead of the usual *stochastic gradient descent* (SGD).

NNGP performs well

Table 1: The NNGP often outperforms finite width networks. Test accuracy on MNIST and CIFAR-10 datasets. The reported NNGP results correspond to the best performing depth, σ_w^2, and σ_b^2 values on the validation set. The traditional NN results correspond to the best performing depth, width and optimization hyperparameters. Best models for a given training set size are specified by (depth-width-σ_w^2-σ_b^2) for NNs and (depth–σ_w^2-σ_b^2) for GPs. More results are in Appendix Table 2.

Num training	Model (ReLU)	Test accuracy	Model (tanh)	Test accuracy
MNIST:1k	NN-2-5000-3.19-0.00	0.9252	NN-2-1000-0.60-0.00	0.9254
	GP-20-1.45-0.28	**0.9279**	GP-20-1.96-0.62	0.9266
MNIST:10k	NN-2-2000-0.42-0.16	0.9771	NN-2-2000-2.41-1.84	0.9745
	GP-7-0.61-0.07	0.9765	GP-2-1.62-0.28	**0.9773**
MNIST:50k	NN-2-2000-0.60-0.44	0.9864	NN-2-5000-0.28-0.34	0.9857
	GP-1-0.10-0.48	0.9875	GP-1-1.28-0.00	**0.9879**
CIFAR:1k	NN-5-500-1.29-0.28	0.3225	NN-1-200-1.45-0.12	0.3378
	GP-7-1.28-0.00	0.3608	GP-50-2.97-0.97	**0.3702**
CIFAR:10k	NN-5-2000-1.60-1.07	0.4545	NN-1-500-1.48-1.59	0.4429
	GP-5-2.97-0.28	**0.4780**	GP-7-3.48-2.00	0.4766
CIFAR:45k	NN-3-5000-0.53-0.01	0.5313	NN-2-2000-1.05-2.08	0.5034
	GP-3-3.31-1.86	**0.5566**	GP-3-3.48-1.52	0.5558

CNNs

- Convolutional neural networks (CNNs) are suitable for learning from dependent tensor data, like images, spatio-temporal matrices and tensors, multi-channel images, and so on.
- (Unlike MLPs), CNNS typically use low-dimensional parameters, including a *kernel* filters, along with some other parameters like strides, pooling parameters and biases.
- An infinite width/depth is not a very realistic scenario for CNNs.

GPs in CNNs

- Main idea due to Garriga-Alonso *et al.*, ICLR, 2019. (Major over-simplification below.)
- Cleverly rewrite a CNN using positions of the kernel filter weights. Use independent Gaussians for filter weights and biases.
- Certain covariances can be ignored, so only variances are involved in the kernel computations
- Kernel recursions become remarkably simple, and can be efficiently computed.

CNN-GP performs well

Method	#samples	Validation error	Test error
NNGP (Lee et al., 2017)	≈ 250	–	1.21%
Convolutional GP (van der Wilk et al., 2017)	SGD	–	1.17%
Deep Conv. GP (Kumar et al., 2018)	SGD	–	1.34%
ConvNet GP	27	0.71%	1.03%
Residual CNN GP	27	0.71%	0.93%
ResNet GP	–	0.68%	**0.84%**
GP + parametric deep kernel (Bradshaw et al., 2017)	SGD	–	0.60%
ResNet (Chen et al., 2018)	–	–	**0.41%**

Table 1: MNIST classification results. #samples gives the number of kernels that were randomly sampled for the hyperparameter search. "ConvNet GP" and "Residual CNN GP" are random CNN architectures with a fixed filter size, whereas "ResNet GP" is a slight modification of the architecture by He et al. (2016b). Entries labelled "SGD" used stochastic gradient descent for tuning hyperparameters, by maximising the likelihood of the training set. The last two methods use parametric neural networks. The hyperparameters of the ResNet GP were not optimised (they were fixed based on the architecture from He et al., 2016b). See Table 2 (appendix) for optimised hyperparameter values.

Acknowledgment:

This research is partially supported by the National Science Foundation (NSF) under grants # DMS-1737918, # OAC-1939916 and # DMR-1939956.

Thank you

References I

[1] Thang Bui, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep Gaussian processes for regression using approximate expectation propagation. In *International Conference on Machine Learning*, pages 1472–1481, 2016.

[2] Youngmin Cho and Lawrence K Saul. Kernel methods for deep learning. In *Advances in neural information processing systems*, pages 342–350, 2009.

[3] Andreas Damianou and Neil Lawrence. Deep Gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

[4] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050 – 1059, 2016.

[5] Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep convolutional networks as shallow Gaussian processes. *arXiv preprint arXiv:1808.05587*, 2018.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.

References II

[7] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571 – 8580, 2018.

[8] Mohammad Emtiyaz E Khan, Alexander Immer, Ehsan Abedi, and Maciej Korzepa. Approximate inference turns deep networks into Gaussian processes. In *Advances In Neural Information Processing Systems*, pages 3094–3104, 2019.

[9] Vinayak Kumar, Vaibhav Singh, PK Srijith, and Andreas Damianou. Deep Gaussian processes with convolutional kernels. *arXiv preprint arXiv:1806.01655*, 2018.

[10] Nicolas Le Roux and Yoshua Bengio. Deep belief networks are compact universal approximators. *Neural Computation*, 22(8):2192 – 2207, 2010.

[11] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[12] Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. VAMPnets for deep learning of molecular kinetics. *Nature Communications*, 9(5):1–11, 2018.

References III

[13] Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

[14] Radford M Neal. *Bayesian Learning for Neural Networks*, volume 118. Springer Science & Business Media, 2012.

[15] Ilya Sutskever and Geoffrey E Hinton. Deep, narrow sigmoid belief networks are universal approximators. *Neural computation*, 20(11):2629 – 2636, 2008.

[16] Mark Van der Wilk, Carl Edward Rasmussen, and James Hensman. Convolutional Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 2849 – 2858, 2017.

[17] Hao Wang and Dit-Yan Yeung. Towards Bayesian deep learning: A survey. *arXiv preprint arXiv:1604.01662*, 2016.

[18] Christopher K. I. Williams. Computing with infinite networks. In *Advances in Neural Information Processing Systems*, pages 295–301, 1997.

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

[20] Hao Wu, Fabian Paul, Christoph Wehmeyer, and Frank Noé. Multiensemble Markov models of molecular thermodynamics and kinetics. *Proceedings of the National Academy of Sciences*, 113(23):E3221–E3230, 2016.