



AUGUSTUS

Accelerating Gaussian Processes and Deep Kernel Networks on GPUs

Alexander Lavin

NVIDIA GTC 2020, Startup Showcase

Gaussian Processes

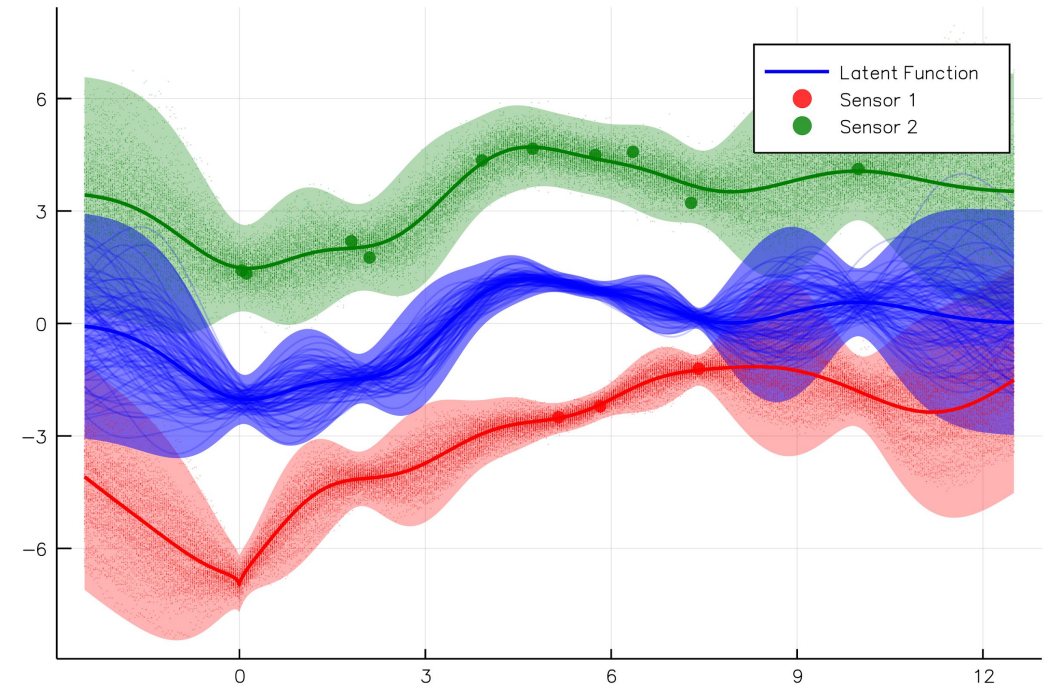
A **Gaussian Process (GP)** defines the distribution over the possible functions $f(x)$ that are consistent with the observed data X .

Why GP?

- Very flexible, Bayesian nonparametric model for unknown functions
- Powerful for regression, classification, unsupervised learning, and other applications that require inference on functions
- **Uncertainty reasoning and interpretability for free**

Widely used across academia and industry

- Applications from disease modeling to climate predictions to financial modeling to robotics control...
- Well-supported libraries: GPyTorch, GPFlow, Stheno.jl, etc.



Example of using GPs for modeling noisy machine sensors.

Using Gaussian Process Models

The *Automatic Statistician*¹ is a tool that uses GPs and flexible kernels to automatically discover plausible models from time-series data.

Bayesian Optimization (BO): a class of approaches to black-box function optimization that typically utilizes a GP surrogate to model an expensive objective.

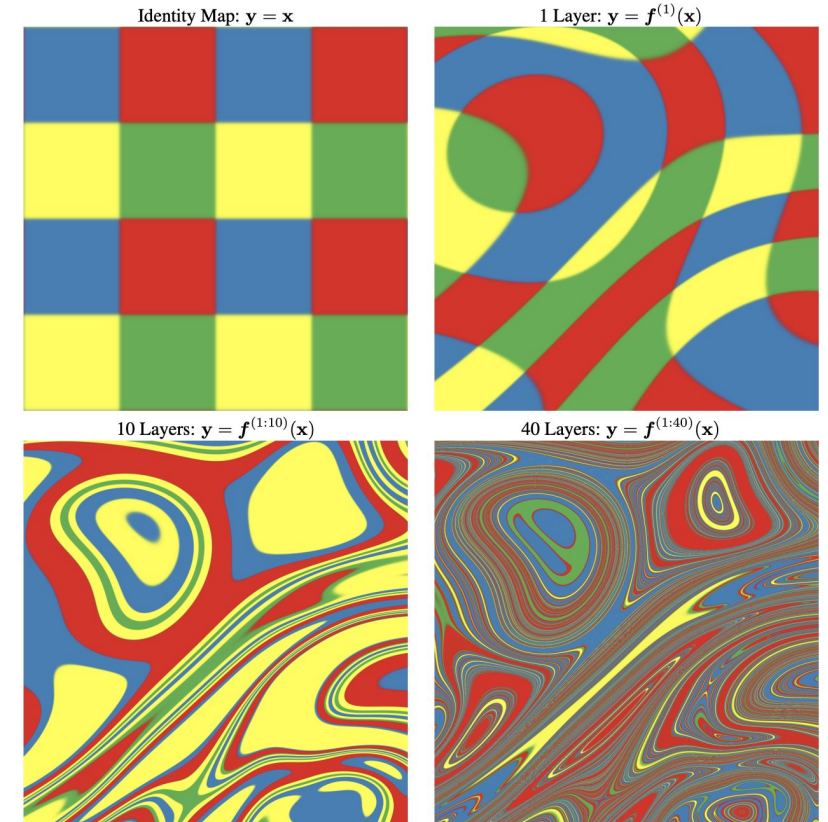
Deep-GP: deep neural networks as compositions of functions drawn from GP priors (right).

Models with GP-priors:

- GPPVAE³ to better model correlations in time-series data
- GPMP⁴ frames robot motion planning as probabilistic inference

But in general $O(N^3)$ inference and $O(N^2)$ space

... scaling is the common issue preventing wider utility!



Visualization of mapping of a two dimensional space through a deep Gaussian Process.²

[1] Zoubin Ghahramani et al.: automaticstatistician.com

[2] Duvenaud et al. (2016) Avoiding pathologies in very deep networks: arxiv.org/abs/1402.5836

[3] Casale et al. Gaussian Process Prior Variational Autoencoders. NeurIPS 2018.

[4] Dong et al. Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs. RSS 2016.

How to accelerate and scale Gaussian Processes?

Sparse GPs are typically used in practice¹:

- Approximate the GP with m inducing points, yielding $O(nm^2)$ inference and $O(nm)$ storage
- Variational methods optimally select inducing points
- Stochastic variational inference methods allows GPs to be fitted to millions of data with $O(m^3)$

Can we utilize advances in ML hardware?

Deep learning can parallelize massive matrix computations on GPU, but [the GP bottleneck is a different matrix operation, the Cholesky decomposition...](#)

[1] Recommended papers on approximate / scaling GPs:

- Titsias. Variational learning of inducing variables in sparse Gaussian processes. AISTATS, 2009.
- Hensman et al. Gaussian processes for big data. UAI, 2013
- Hensman et al. Scalable variational Gaussian process classification. ICML, 2015.
- Wilson et al. Thoughts on massively scalable Gaussian processes. arXiv:1511.01870, 2015.

Avoiding Cholesky to parallelize GPs on GPU

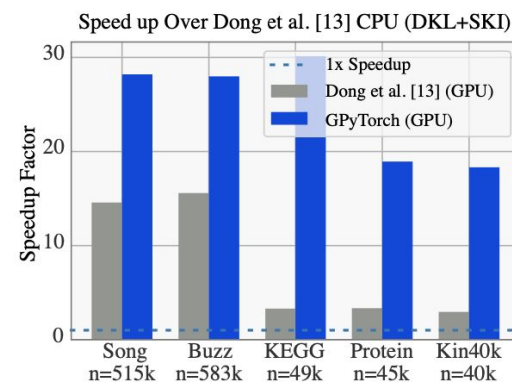
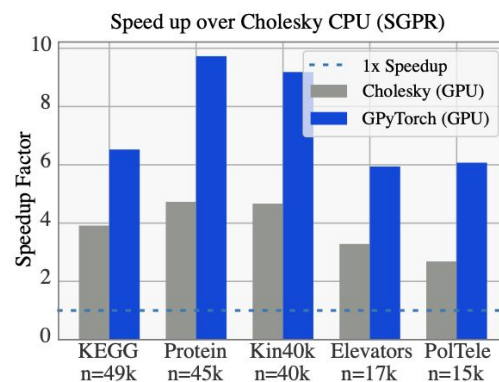
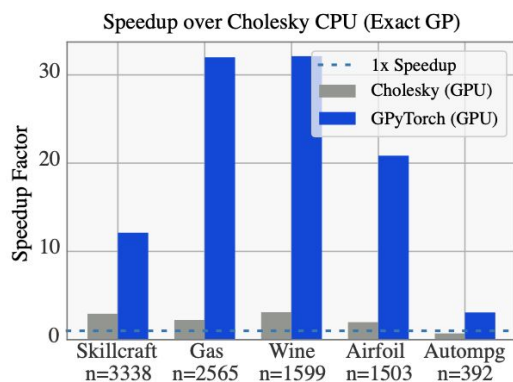
Blackbox Matrix-Matrix (BBMM) method,

- reduces the bulk of GP inference to highly-parallelizable matrix-matrix multiplication.
- reduces the time complexity of **exact GP inference** from $O(n^3)$ to $O(n^2)$.

Standard in GPyTorch library!

GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration

Jacob R. Gardner*, Geoff Pleiss*,
David Bindel, Kilian Q. Weinberger, Andrew Gordon Wilson
Cornell University
{jrg365, kqw4, andrew}@cornell.edu,
{geoff, bindel}@cs.cornell.edu



[1] Garner et al. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. NeurIPS 2018.

Challenging use-case: Deep Kernel Nets for Computer Vision

*Deep Kernel Learning (DKL)*¹ combines the structural properties of neural networks as feature-extractors, with the non-parametric flexibility of kernel methods (i.e. Gaussian processes).

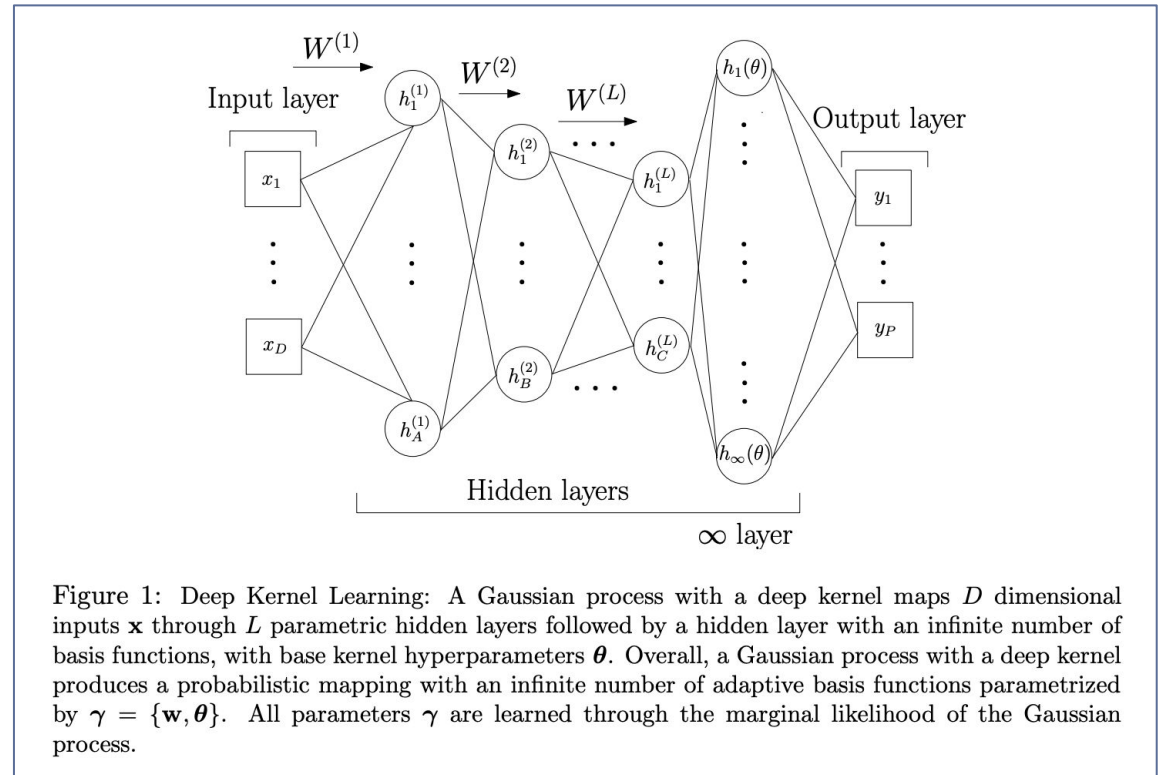
For computer vision we can use:

1. ConvNet feature extractor (e.g. DenseNet)
2. GP regression layer on top

Yields near state-of-art on object classification tasks, **with uncertainty reasoning!**

Scalable inference on GPU via BBMM.

Moar? We can implement *probabilistic programmed* version by warping the GP and ConvNet into a new kernel.




[1] Wilson et al. Deep Kernel Learning. AISTATS, 2015.

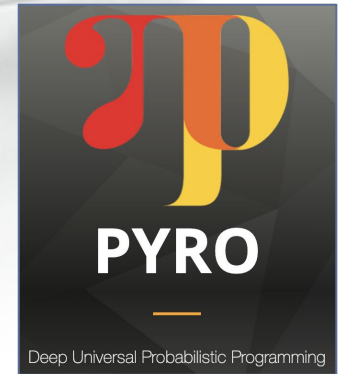
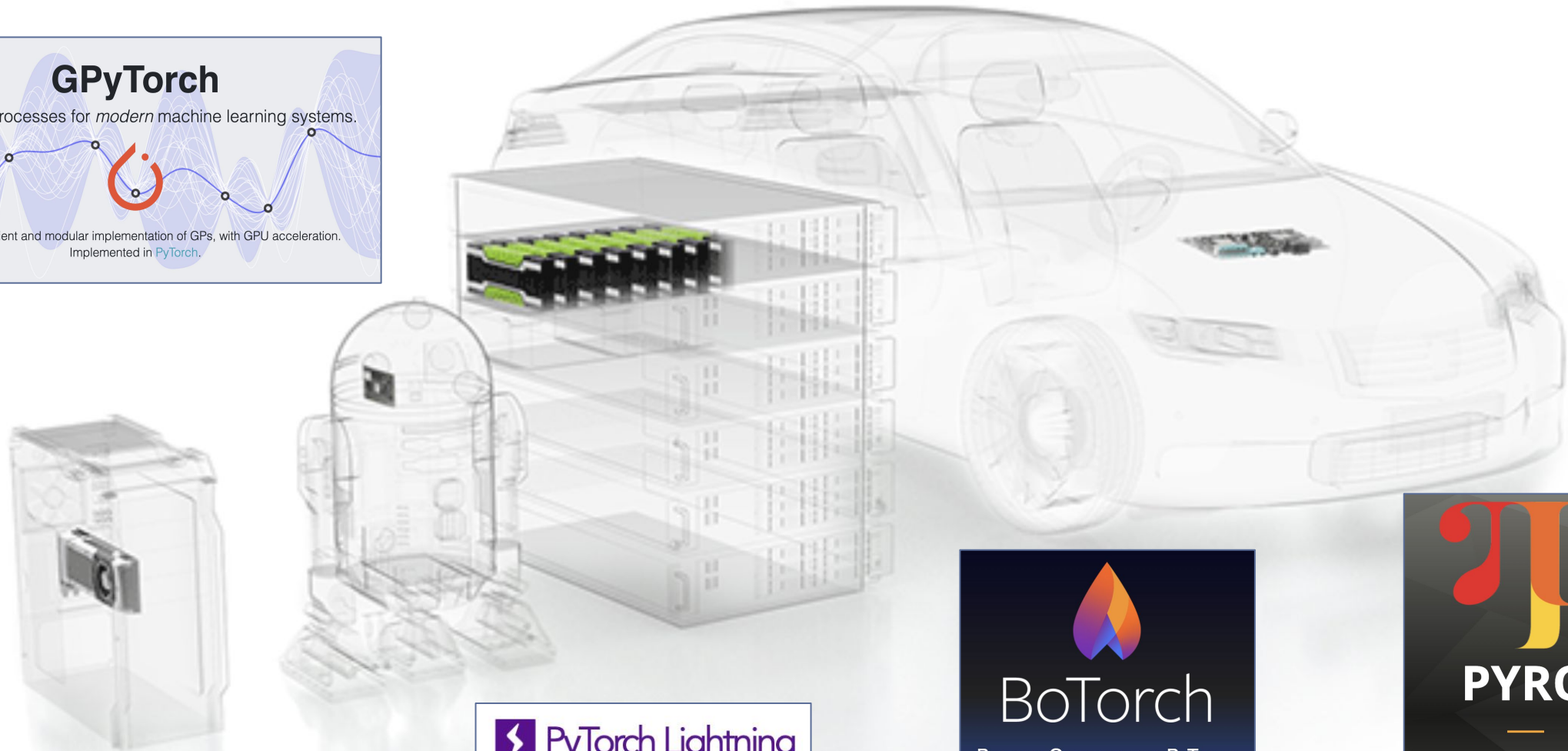
[2] van de Meent et al. (2018) An Introduction to Probabilistic Programming. arxiv.org/abs/1809.10756.

PyTorch + NVIDIA GPU Ecosystem makes this possible, *and useful in industry settings*

GPyTorch
Gaussian processes for *modern* machine learning systems.



A highly efficient and modular implementation of GPs, with GPU acceleration.
Implemented in *PyTorch*.



(image: blogs.nvidia.com/blog/2016/01/12/accelerating-ai-artificial-intelligence-gpus)

Building human-centric AI at Augustus Intelligence

Augustus Intelligence

At **Augustus Intelligence** we seek to provide intelligent, decision-making systems where others cannot. This calls for accelerating existing and state-of-the-art AI & ML towards production systems, while innovating in novel, creative ways. The team at Augustus develops fundamental advances in AI, and repeatedly turns breakthrough technology ideas into products towards solving real-world problems with massive impact.

With an international team of experts -- SF to NYC to Paris -- we develop and deploy trusted enterprise products that drive human-machine synergies across industries.

Augustus AI leadership:

Louis Monier, Chief Scientific Officer

AltaVista co-founder, Xerox PARC researcher, eBay Fellow, Airbnb Head of AI, and more.

Gregory Renard, Chief AI Officer

Renowned expert in NLP and knowledge graphs, decades of experience deploying augmented-intelligence products.

Alexander Lavin, Technical Fellow & Director of Research

Former rocket scientist turned AI research engineer, leading expert in Bayesian machine learning.

Thank You

e: alexander.lavin@augustusai.com

t: [@theAlexLavin](#)

w: lavin.io

