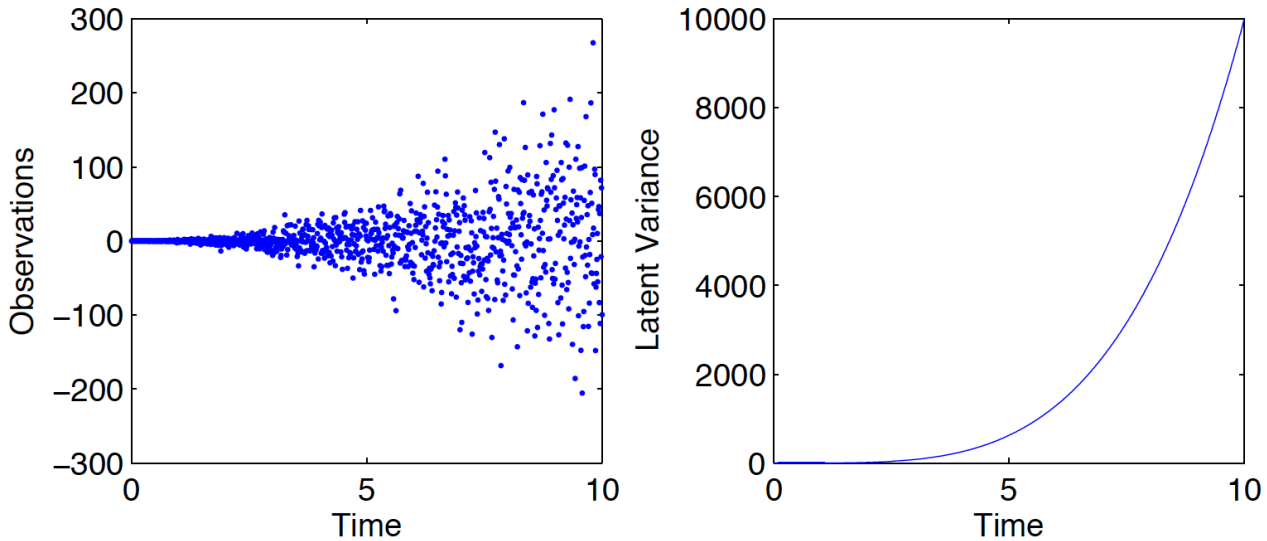
Bayesian Neural Networks from a Gaussian Process Perspective

Andrew Gordon Wilson

https://cims.nyu.edu/~andrewgw
Courant Institute of Mathematical Sciences
Center for Data Science
New York University

Gaussian Process Summer School
September 16, 2020

Last Time... Machine Learning for Econometrics (The Start of My Journey...)



Autoregressive Conditional Heteroscedasticity (ARCH)

2003 Nobel Prize in Economics

$$y(t) = \mathcal{N}(y(t); 0, a_0 + a_1 y(t-1)^2)$$

Autoregressive Conditional Heteroscedasticity (ARCH)
2003 Nobel Prize in Economics

$$y(t) = \mathcal{N}(y(t); 0, a_0 + a_1 y(t-1)^2)$$

Gaussian Copula Process Volatility (GCPV)
(My First PhD Project)

$$y(x) = \mathcal{N}(y(x); 0, f(x)^2)$$
$$f(x) \sim \mathcal{GP}(m(x), k(x, x'))$$

- Can approximate a much greater range of variance functions
- Operates on continuous inputs x
- Can effortlessly handle missing data
- Can effortlessly accommodate multivariate inputs x (covariates other than time)
- **Observation: performance extremely sensitive to even small changes in kernel hyperparameters**

Heteroscedasticity revisited...

Which of these models do you prefer, and why?

Choice 1

$$y(x)|f(x), g(x) \sim \mathcal{N}(y(x); f(x), g(x)^2)$$
$$f(x) \sim \mathcal{GP}, g(x) \sim \mathcal{GP}$$

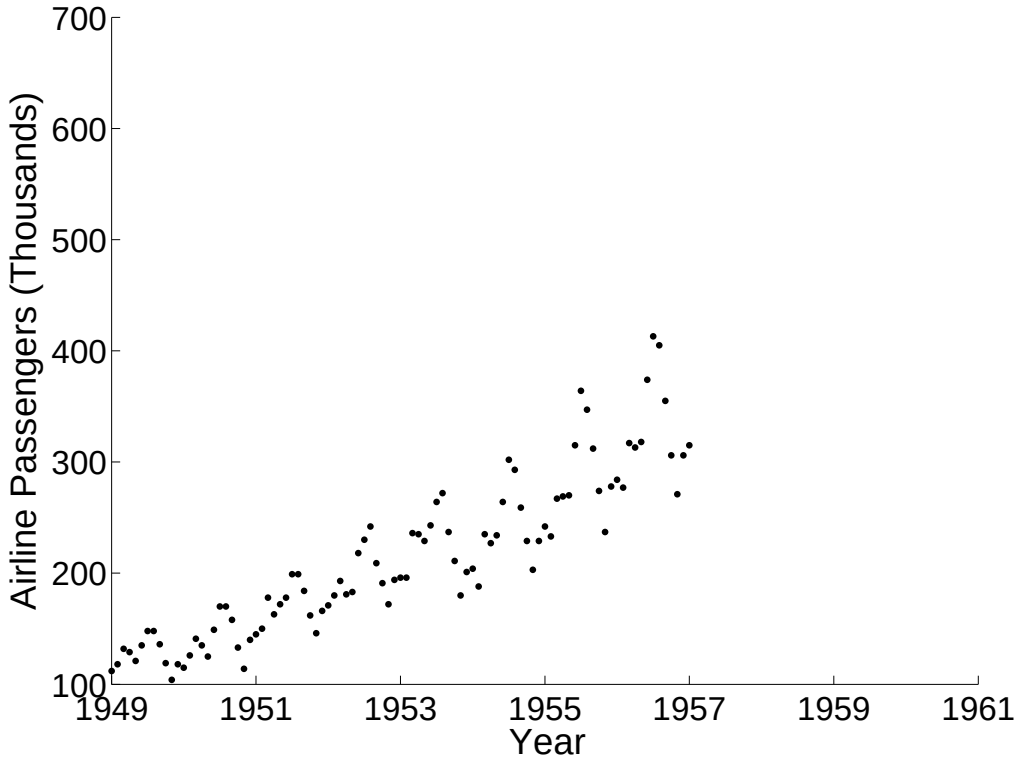
Choice 2

$$y(x)|f(x), g(x) \sim \mathcal{N}(y(x); f(x)g(x), g(x)^2)$$
$$f(x) \sim \mathcal{GP}, g(x) \sim \mathcal{GP}$$

Some conclusions...

- Flexibility isn't the whole story, inductive biases are at least as important.
- Degenerate model specification can be *helpful*, rather than something to necessarily avoid.
- Asymptotic results often mean very little. Rates of convergence, or even intuitions about non-asymptotic behaviour, are more meaningful.
- Infinite models (models with unbounded capacity) are almost always desirable, but the details matter.
- Releasing good code is crucial.
- Try to keep the approach as simple as possible.
- Empirical results often provide the most effective argument.

Model Selection



Which model should we choose?

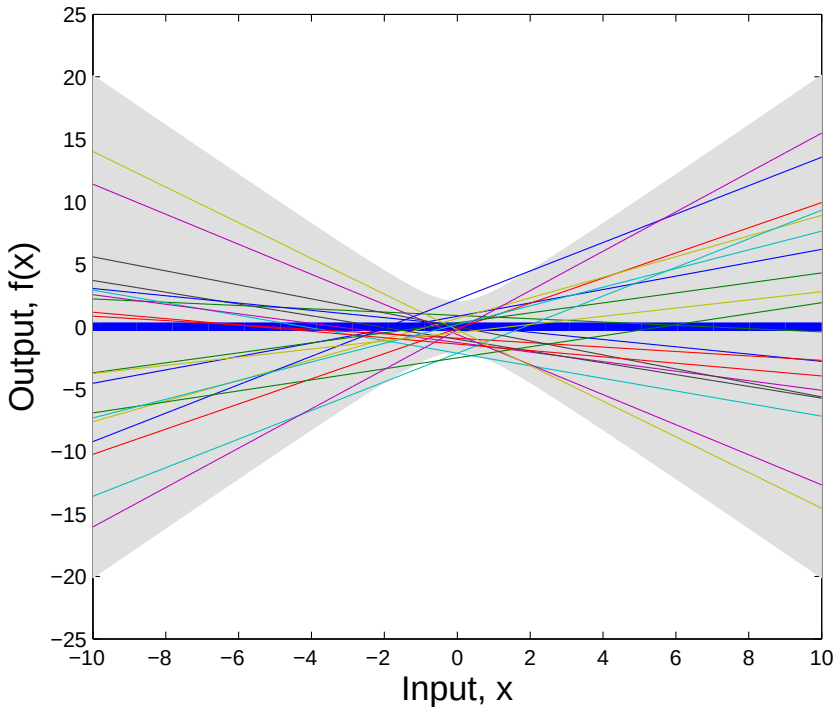
(1): $f_1(x) = w_0 + w_1 x$ **(2):** $f_2(x) = \sum_{j=0}^{3} w_j x^j$ **(3):** $f_3(x) = \sum_{j=0}^{10^4} w_j x^j$

A Function-Space View

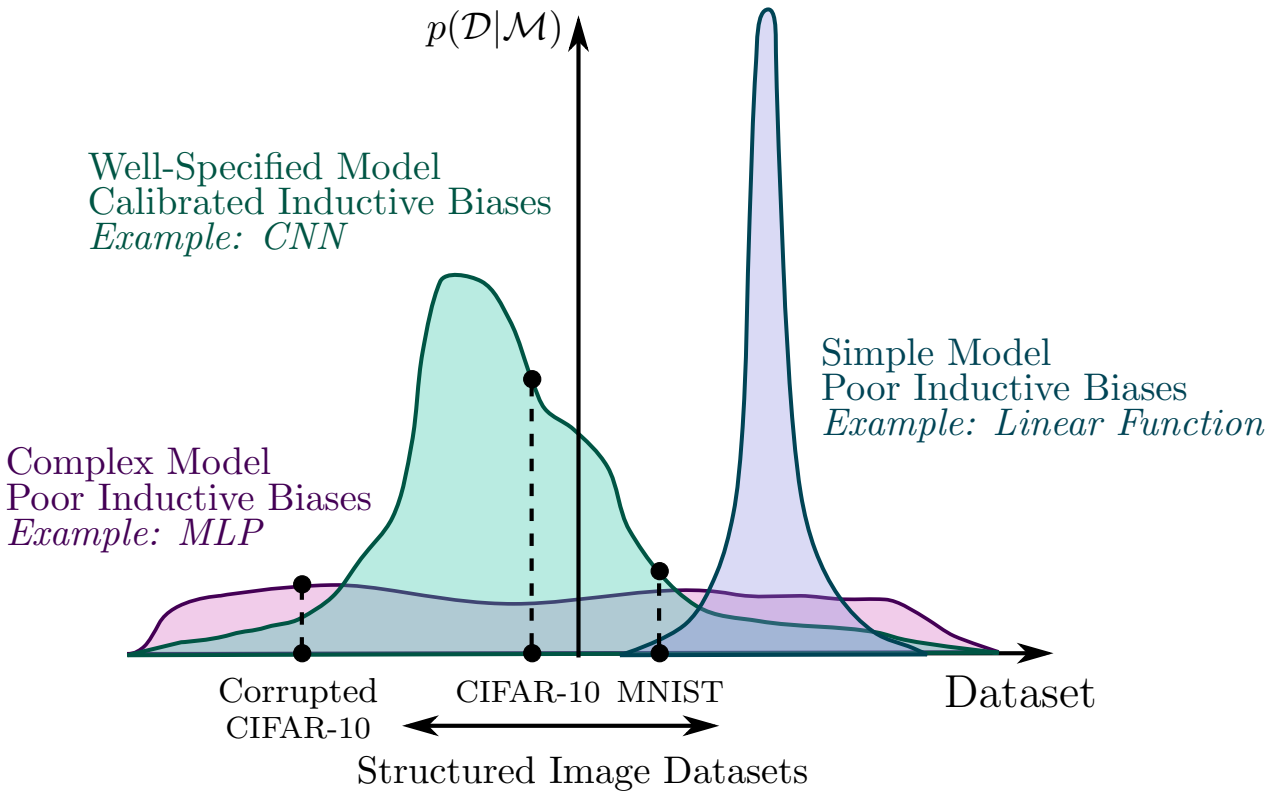
Consider the simple linear model,

$$f(x) = w_0 + w_1 x\,, \tag{1}$$

$$w_0, w_1 \sim \mathcal{N}(0, 1)\,. \tag{2}$$

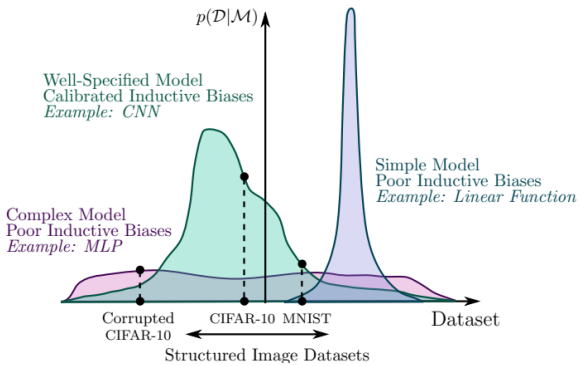


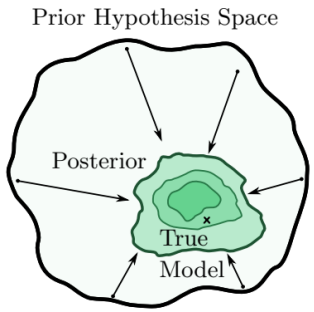
Model Construction and Generalization

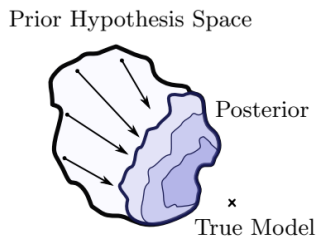


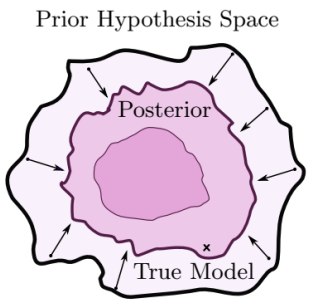
How do we learn?

- The ability for a system to learn is determined by its *support* (which solutions are a priori possible) and *inductive biases* (which solutions are a priori likely).
- We should not conflate *flexibility* and *complexity*.
- An influx of new *massive* datasets provide great opportunities to automatically learn rich statistical structure, leading to new scientific discoveries.





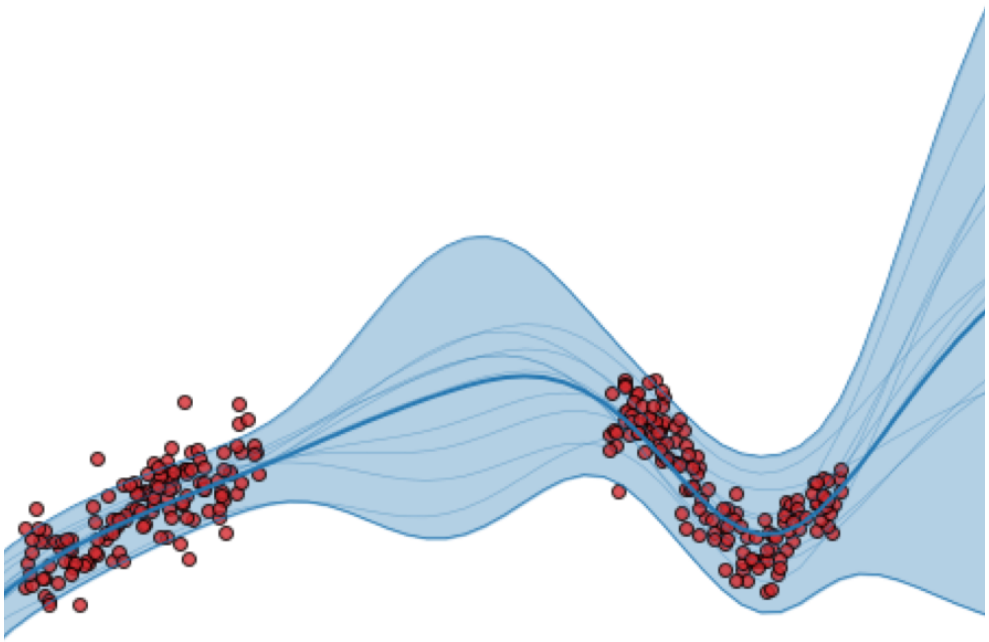




Bayesian Deep Learning and a Probabilistic Perspective of Generalization
Wilson and Izmailov, 2020
arXiv 2002.08791

What is Bayesian learning?

- The key distinguishing property of a Bayesian approach is **marginalization** instead of optimization.
- Rather than use a single setting of parameters $\mathbf{w}$, use all settings weighted by their posterior probabilities in a *Bayesian model average*.



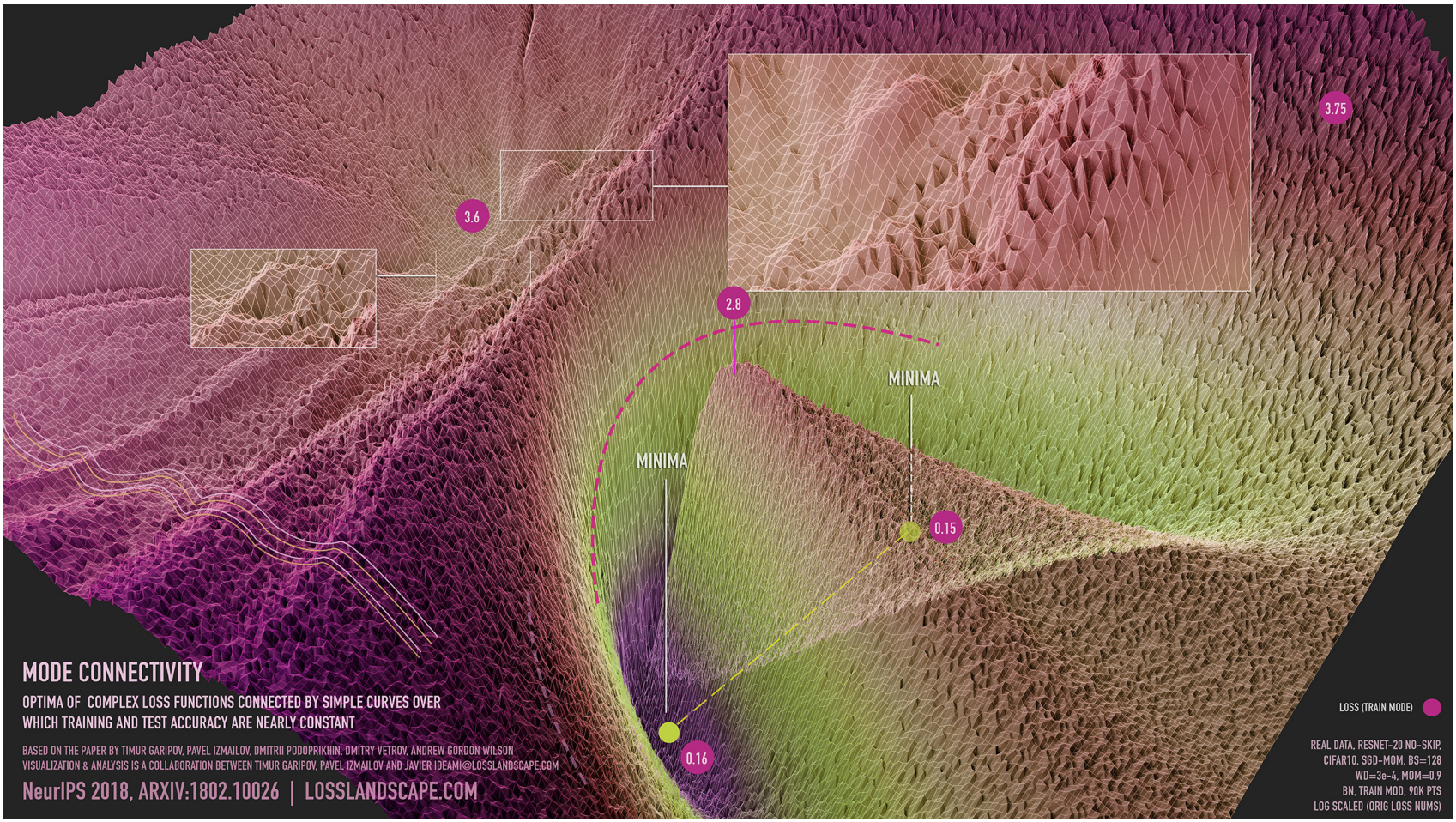
Why Bayesian Deep Learning?

Recall the *Bayesian model average* (BMA):

$$p(y|x_*, \mathcal{D}) = \int p(y|x_*, w)p(w|\mathcal{D})dw\,. \tag{3}$$

- Think of each setting of $\mathbf{w}$ as a different model. Eq. (3) is a *Bayesian model average* over models weighted by their posterior probabilities.
- Represents *epistemic uncertainty* over which $f(x, w)$ fits the data.
- Can view classical training as using an approximate posterior $q(\mathbf{w}|\mathbf{y}, X) = \delta(w = w_{\text{MAP}})$.
- The posterior $p(w|\mathcal{D})$ (or loss $\mathcal{L} = -\log p(w|\mathcal{D})$) for neural networks is extraordinarily complex, containing many complementary solutions, which is why BMA is *especially* significant in deep learning.
- Understanding the structure of neural network loss landscapes is crucial for better estimating the BMA.

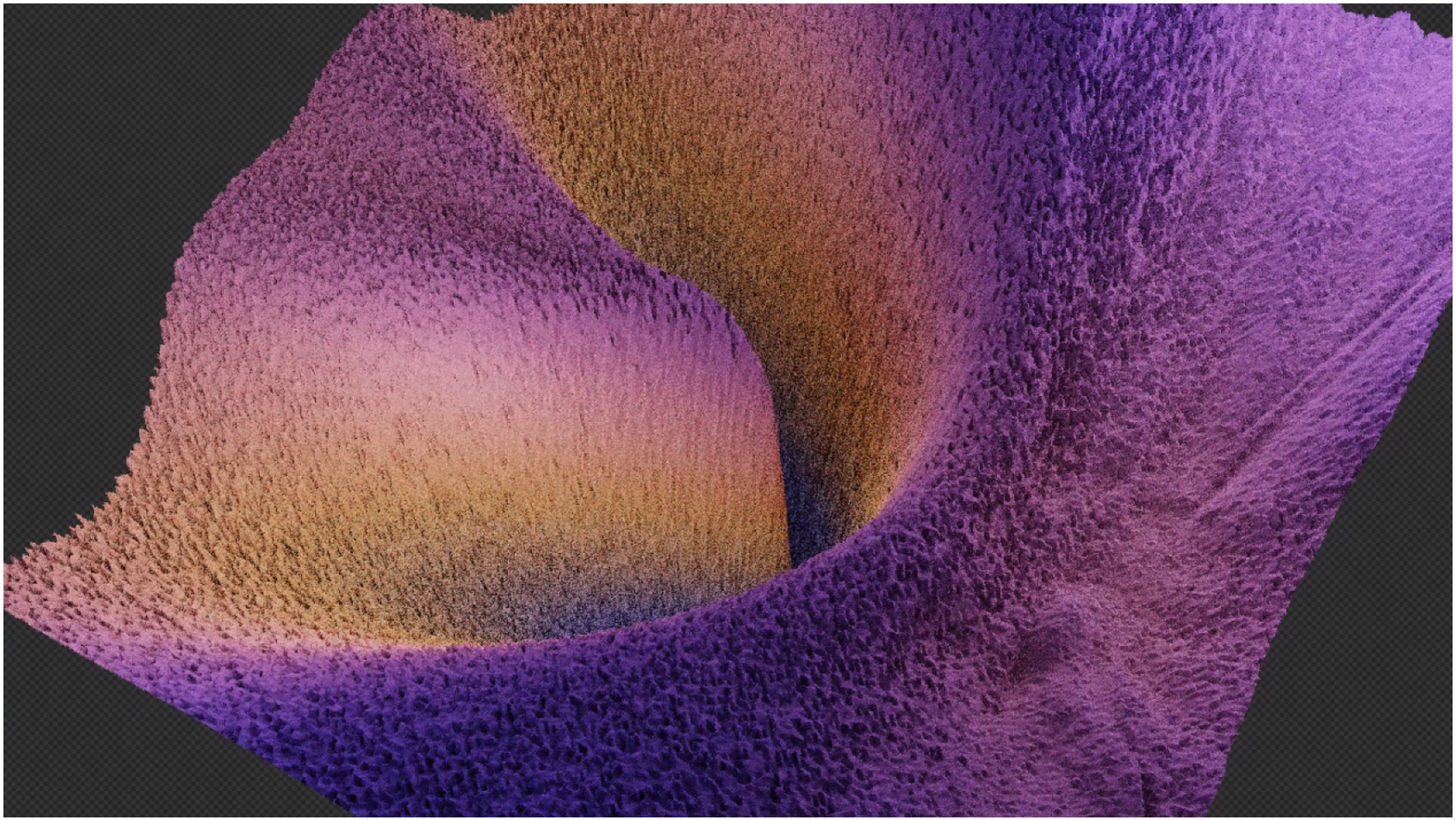
Mode Connectivity



Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs.
T. Garipov, P. Izmailov, D. Podoprikhin, D. Vetrov, A.G. Wilson. NeurIPS 2018.

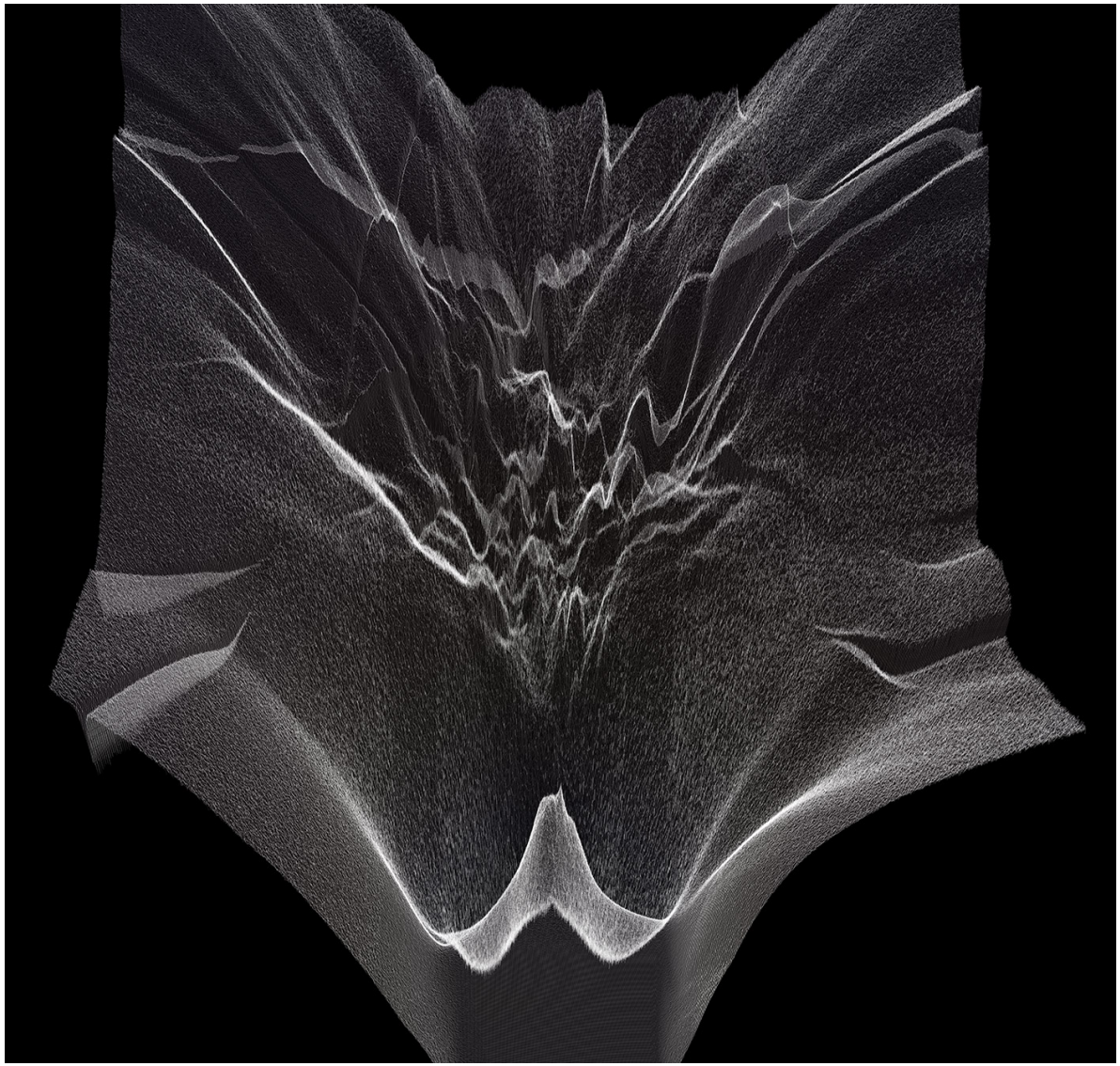
Loss landscape figures in collaboration with Javier Ideami (losslandscape.com).

Mode Connectivity

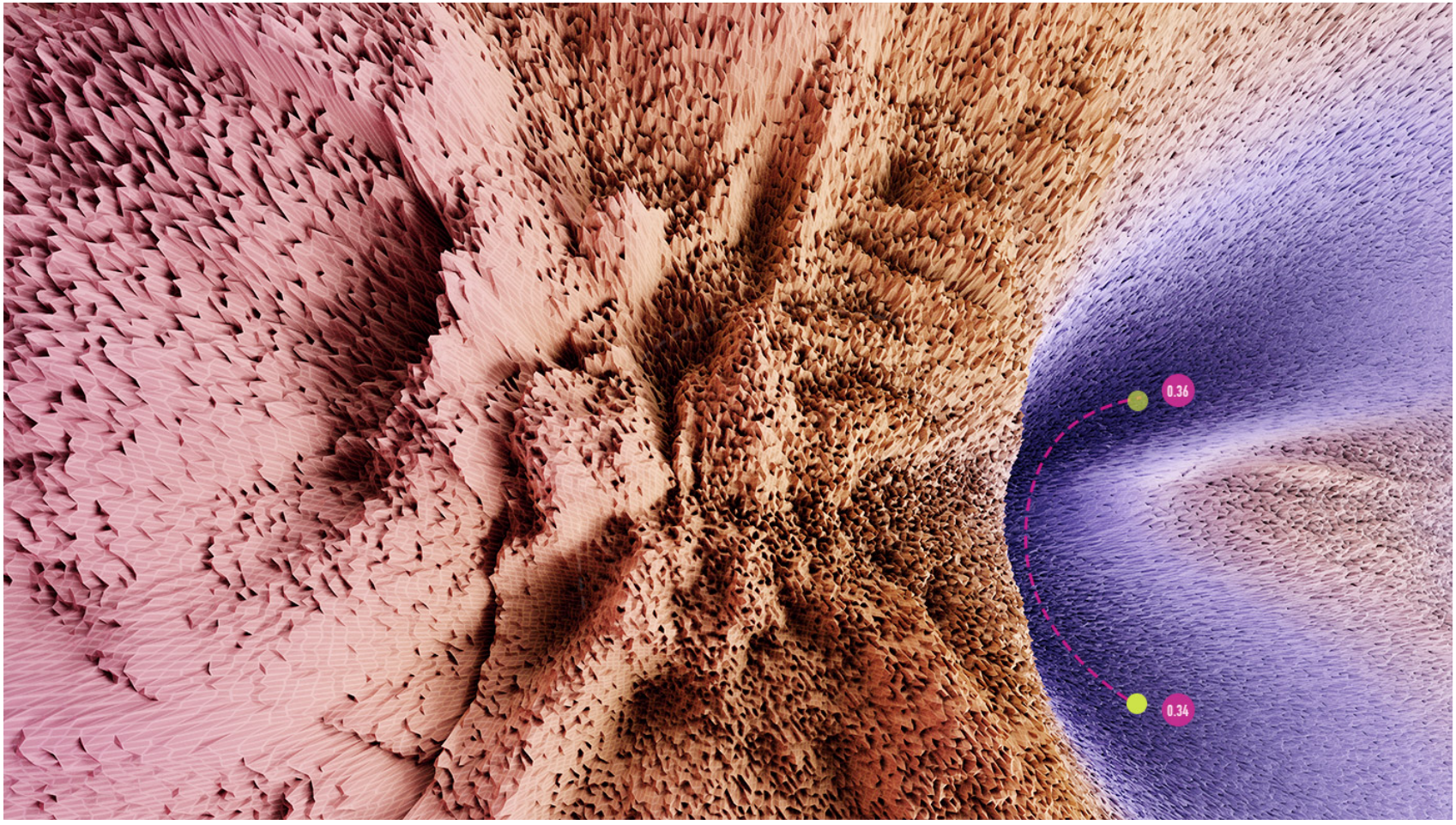


Mode Connectivity

Mode Connectivity

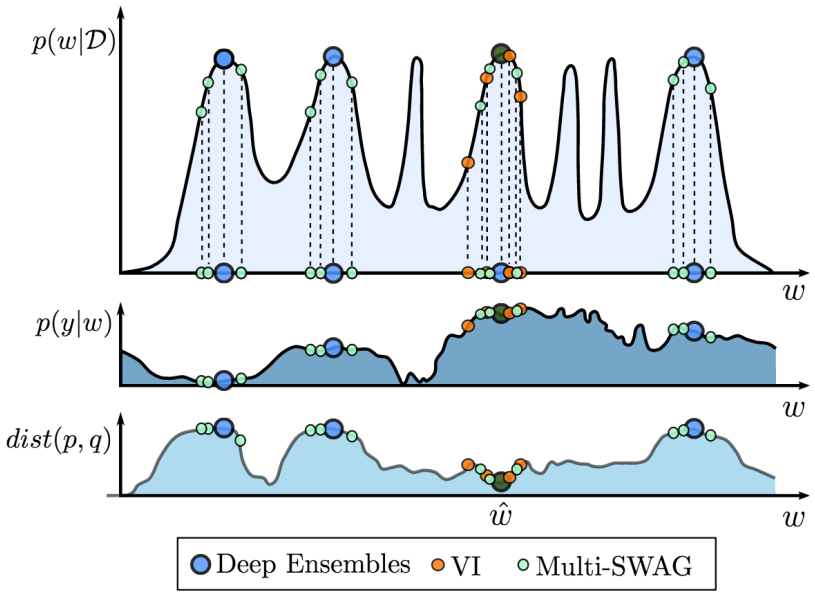


Mode Connectivity



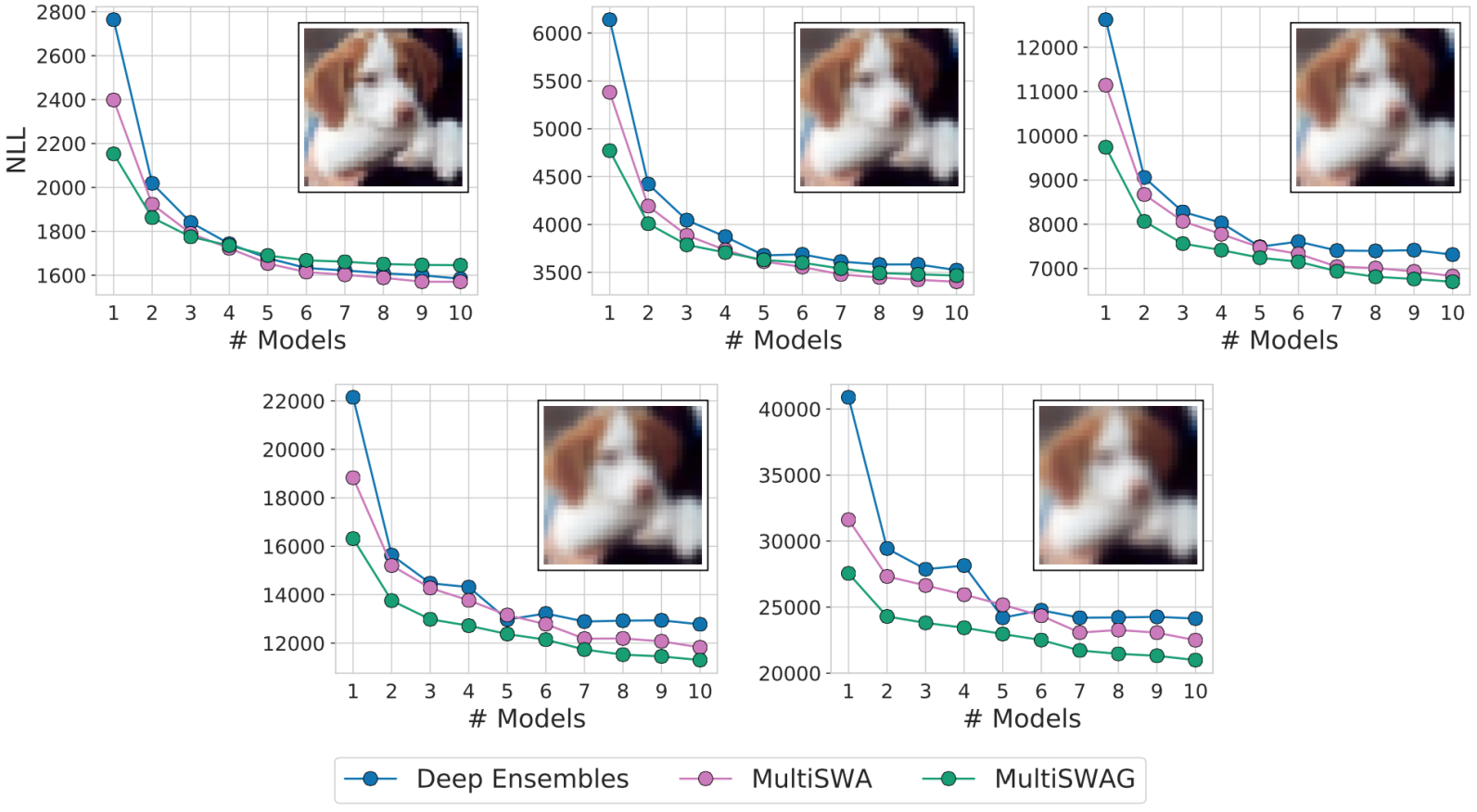
Better Marginalization

$$p(y|x_*, \mathcal{D}) = \int p(y|x_*, w)p(w|\mathcal{D})dw\,. \tag{4}$$



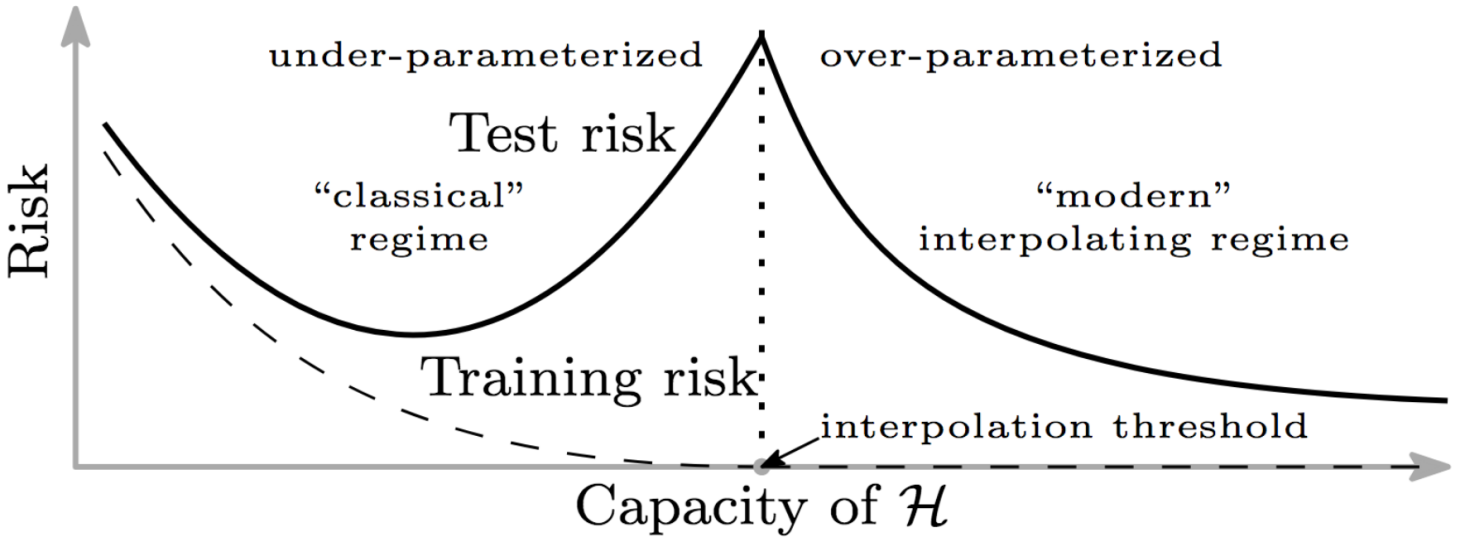
- MultiSWAG forms a Gaussian mixture posterior from multiple independent SWAG solutions.
- Like deep ensembles, MultiSWAG incorporates multiple basins of attraction in the model average, but it additionally marginalizes within basins of attraction for a better approximation to the BMA.

Better Marginalization: MultiSWAG



[1] *Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift.* Ovadia et. al, 2019
[2] *Bayesian Deep Learning and a Probabilistic Perspective of Generalization.* Wilson and Izmailov, 2020

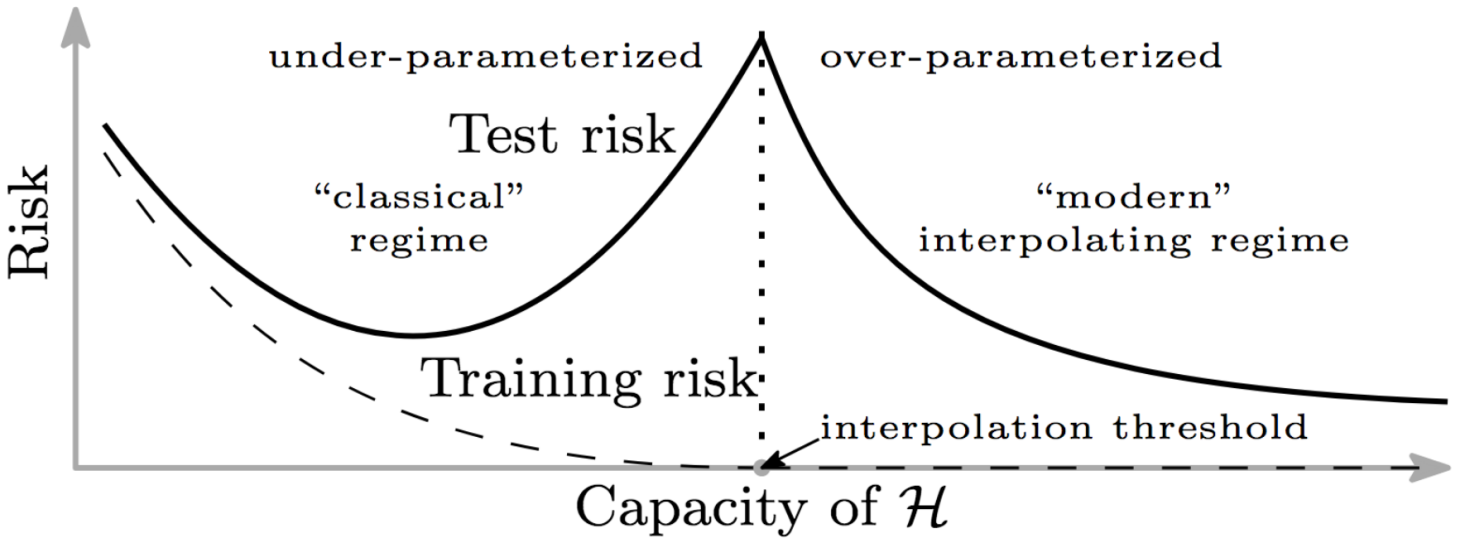
Double Descent



Belkin et. al (2018)

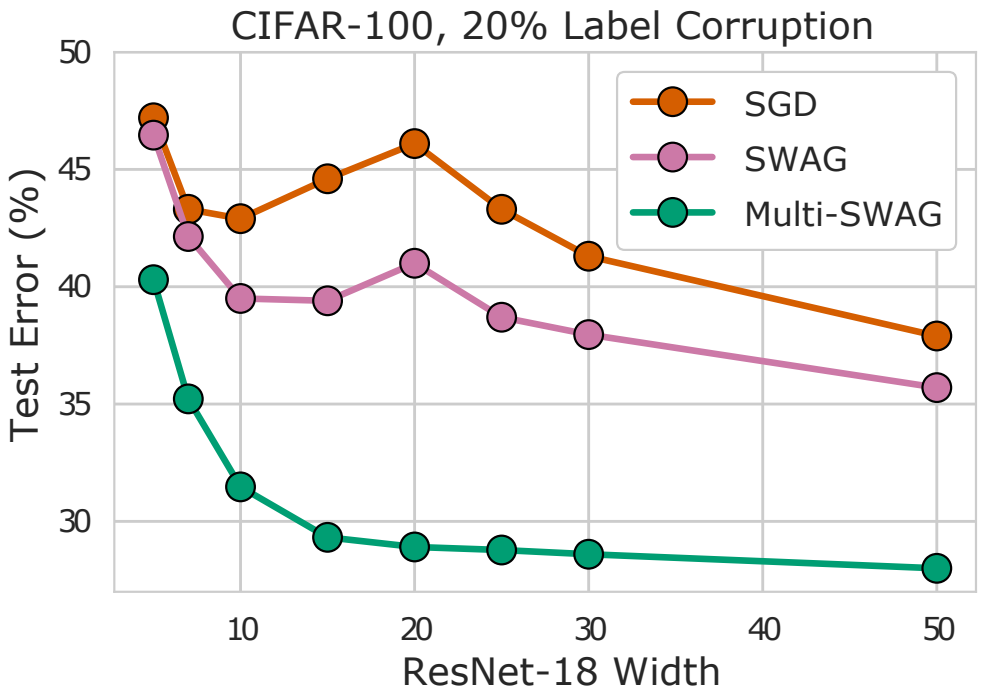
Reconciling modern machine learning practice and the bias-variance trade-off. Belkin et. al, 2018

Double Descent



Should a Bayesian model experience double descent?

Bayesian Model Averaging Alleviates Double Descent



Bayesian Deep Learning and a Probabilistic Perspective of Generalization. Wilson & Izmailov, 2020

Neural Network Priors

A parameter prior $p(w) = \mathcal{N}(0, \alpha^2)$ with a neural network architecture $f(x, w)$ induces a structured distribution over *functions* $p(f(x))$.

Deep Image Prior

- *Randomly initialized* CNNs *without training* provide excellent performance for image denoising, super-resolution, and inpainting: a sample function from $p(f(x))$ captures low-level image statistics, before any training.

Random Network Features

- Pre-processing CIFAR-10 with a *randomly initialized untrained* CNN dramatically improves the test performance of a Gaussian kernel on pixels from 54% accuracy to 71%, with an additional 2% from ℓ_2 regularization.

[1] *Deep Image Prior*. Ulyanov, D., Vedaldi, A., Lempitsky, V. CVPR 2018.
[2] *Understanding Deep Learning Requires Rethinking Generalzation*. Zhang et. al, ICLR 2016.
[3] *Bayesian Deep Learning and a Probabilistic Perspective of Generalization.* Wilson & Izmailov, 2020.

Tempered Posteriors

In Bayesian deep learning it is typical to consider the *tempered* posterior:

$$p_T(w|\mathcal{D}) = \frac{1}{Z(T)} p(\mathcal{D}|w)^{1/T} p(w), \tag{5}$$

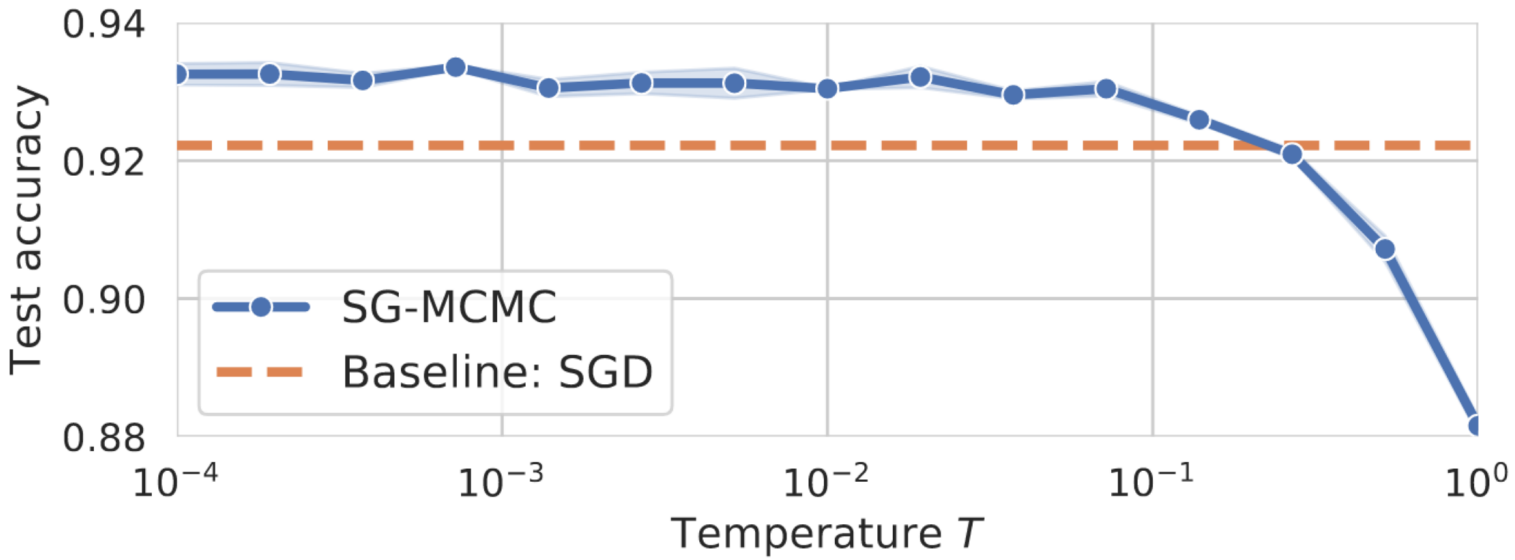
where T is a *temperature* parameter, and $Z(T)$ is the normalizing constant corresponding to temperature T. The temperature parameter controls how the prior and likelihood interact in the posterior:

- $T < 1$ corresponds to *cold posteriors*, where the posterior distribution is more concentrated around solutions with high likelihood.
- $T = 1$ corresponds to the standard Bayesian posterior distribution.
- $T > 1$ corresponds to *warm posteriors*, where the prior effect is stronger and the posterior collapse is slower.

E.g.: *The safe Bayesian*. Grunwald, P. COLT 2012.

Cold Posteriors

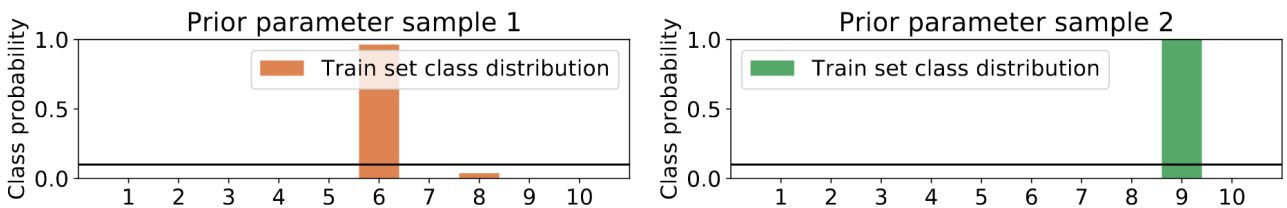
Wenzel et. al (2020) highlight the result that for $p(w) = N(0, I)$ *cold posteriors* with $T < 1$ often provide improved performance.



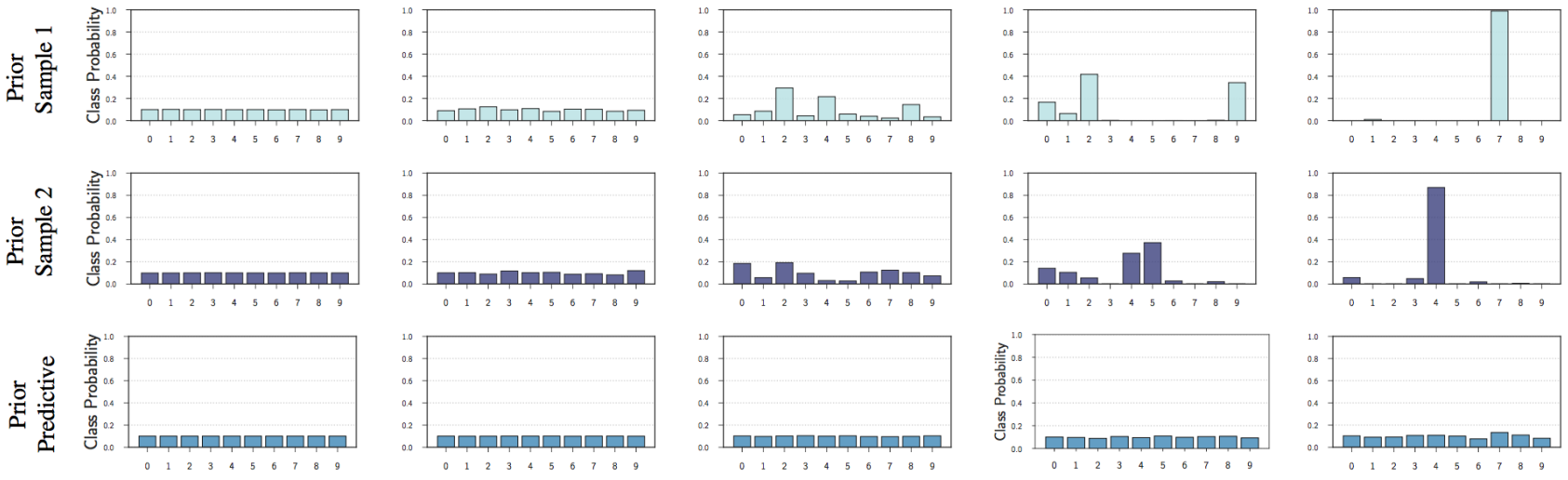
How good is the Bayes posterior in deep neural networks really? Wenzel et. al, ICML 2020.

Prior Misspecification?

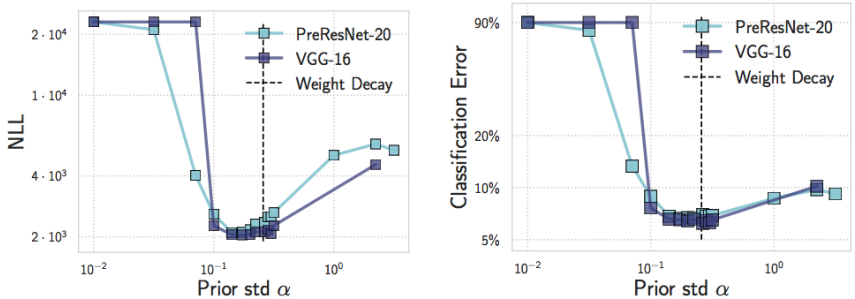
They suggest the result is due to prior misspecification, showing that sample functions $p(f(x))$ seem to assign one label to most classes on CIFAR-10.



Changing the prior variance scale α



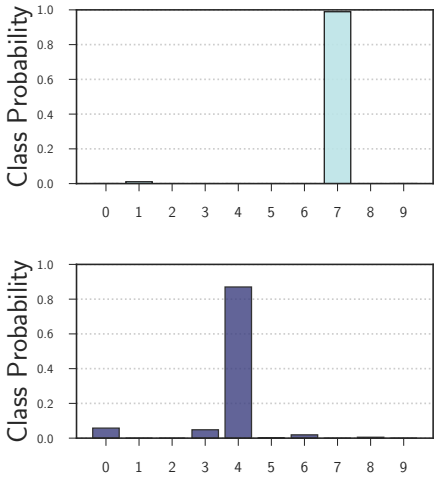
(a) $\alpha = 0.01$ (b) $\alpha = 0.1$ (c) $\alpha = 0.3$ (d) $\alpha = 1.$ (e) $\alpha = \sqrt{10}$



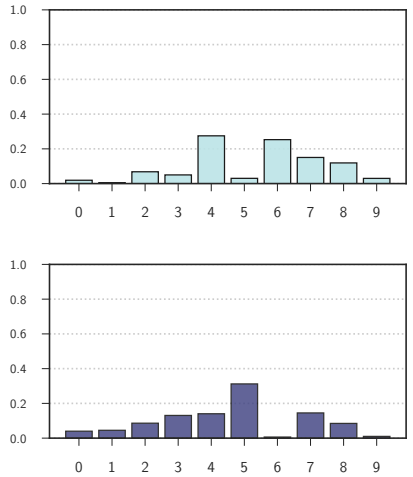
(f) (g)

Bayesian Deep Learning and a Probabilistic Perspective of Generalization. Wilson & Izmailov, 2020.

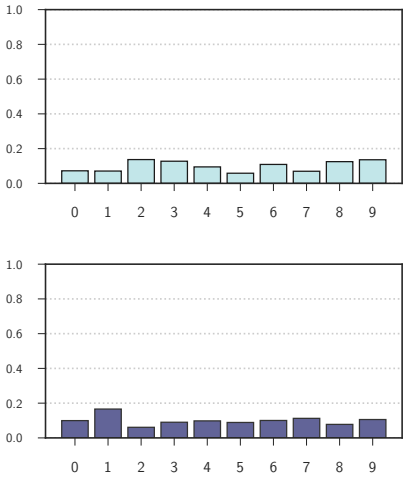
The effect of data on the posterior



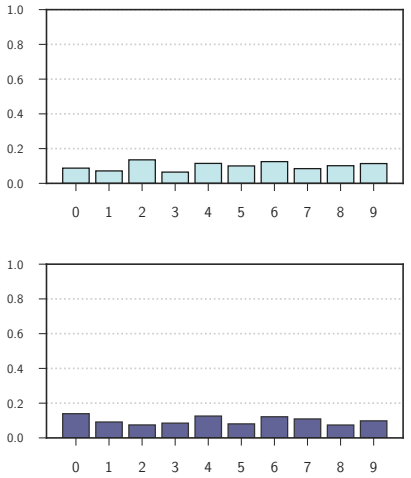
(a) Prior ($\alpha = \sqrt{10}$)



(b) 10 datapoints



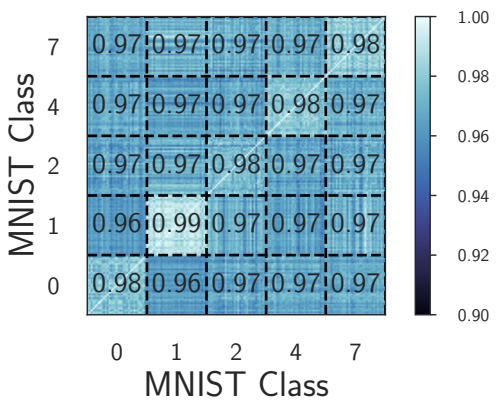
(c) 100 datapoints



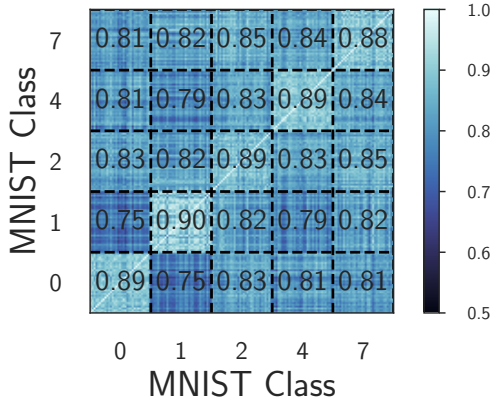
(d) 1000 datapoints

From a Gaussian process perspective, what properties of the prior over functions induced by a Bayesian neural network might you check to see if it seems reasonable?

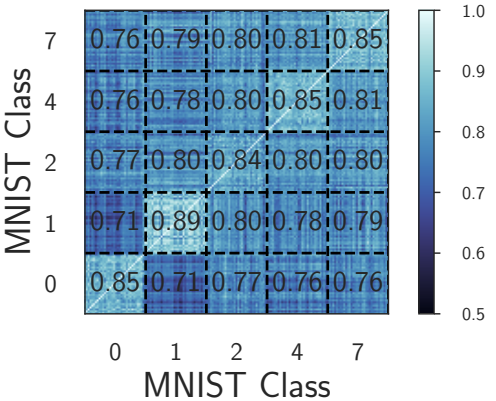
Prior Class Correlations



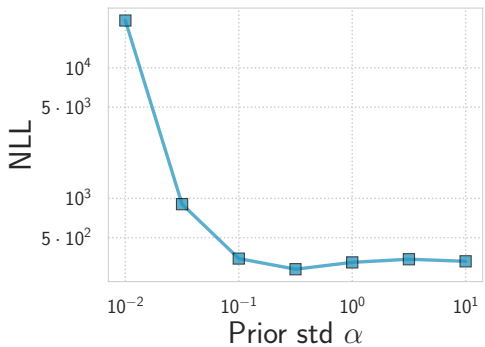
(e) $\alpha = 0.02$



(f) $\alpha = 0.1$



(g) $\alpha = 1$



(h)

Bayesian Deep Learning and a Probabilistic Perspective of Generalization. Wilson & Izmailov, 2020.

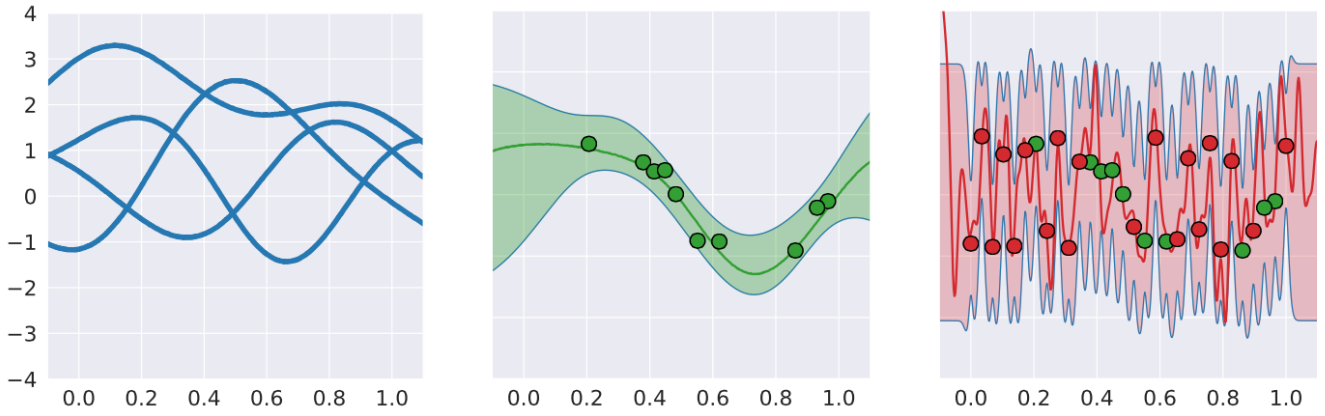
Thoughts on Tempering (Part 1)

- It would be surprising if $T = 1$ was the best setting of this hyperparameter.
- Our models are certainly misspecified, and we should acknowledge that misspecification in our estimation procedure by learning T. Learning T is not too different from learning other properties of the likelihood, such as noise.
- A tempered posterior is a more honest reflection of our prior beliefs than the untempered posterior. Bayesian inference is about honestly reflecting our beliefs in the modelling process.

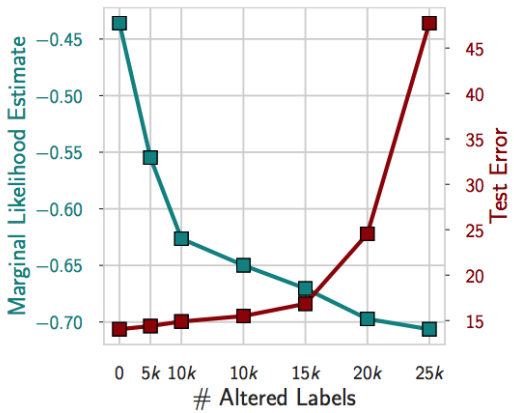
Thoughts on Tempering (Part 2)

- While certainly the prior $p(f(x))$ is misspecified, the result of assigning one class to most data is a *soft* prior bias, which (1) doesn't hurt the predictive distribution, (2) is easily corrected by appropriately setting the prior parameter variance α^2, and (3) is quickly modulated by data.
- More important is the induced *covariance function* (kernel) over images, which appears reasonable. The deep image prior and random network feature results also suggest this prior is largely reasonable.
- In addition to not tuning α, the result in Wenzel et. al (2020) could have been exacerbated due to lack of multimodal marginalization.
- There are cases when $T < 1$ will help given a finite number of samples, even if the untempered model is correctly specified. Imagine estimating the mean of $\mathcal{N}(0, I)$ from samples where $d \gg 1$. The samples will have norm close to $\sqrt{d}$.

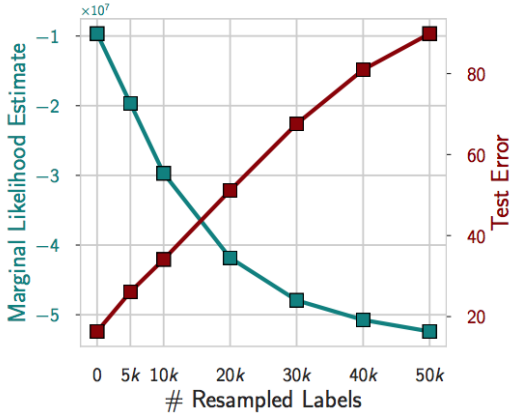
Rethinking Generalization



(a) Prior Draws (b) True Labels (c) Corrupted Labels



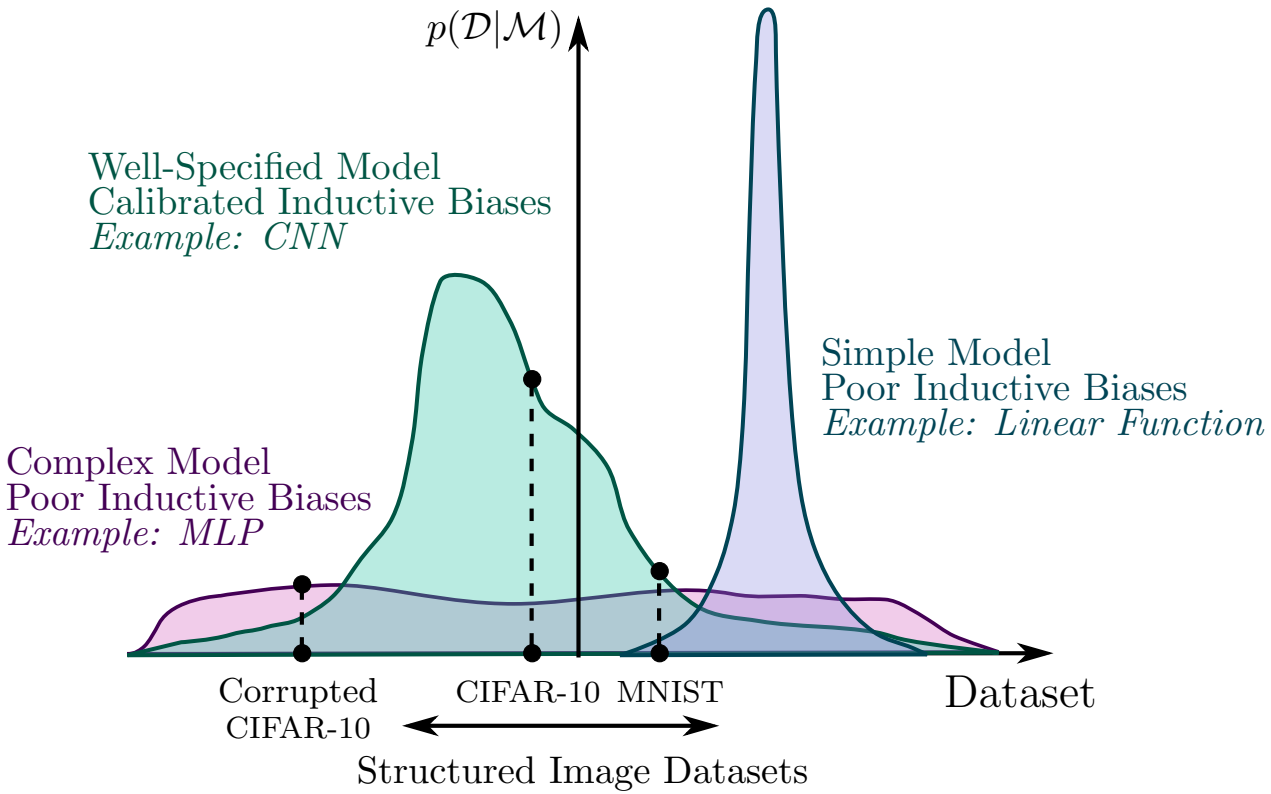
(d) Gaussian Process



(e) PreResNet-20

[1] *Understanding Deep Learning Requires Rethinking Generalzation*. Zhang et. al, ICLR 2016.
[2] *Bayesian Deep Learning and a Probabilistic Perspective of Generalization.* Wilson & Izmailov, 2020.

Model Construction



Function Space Priors

We should embrace the function space perspective in constructing priors.

- However, if we contrive priors over parameters $p(w)$ to induce distributions over functions $p(f)$ that resemble familiar models such as Gaussian processes with RBF kernels, we could be throwing the baby out with the bathwater.
- Indeed, neural networks are useful as their own model class precisely because they have different inductive biases from other models.
- We should try to gain insights by thinking in *function space*, but note that architecture design itself is thinking in function space: properties such as equivariance to translations in convolutional architectures imbue the associated distribution over functions with these properties.

PAC-Bayes

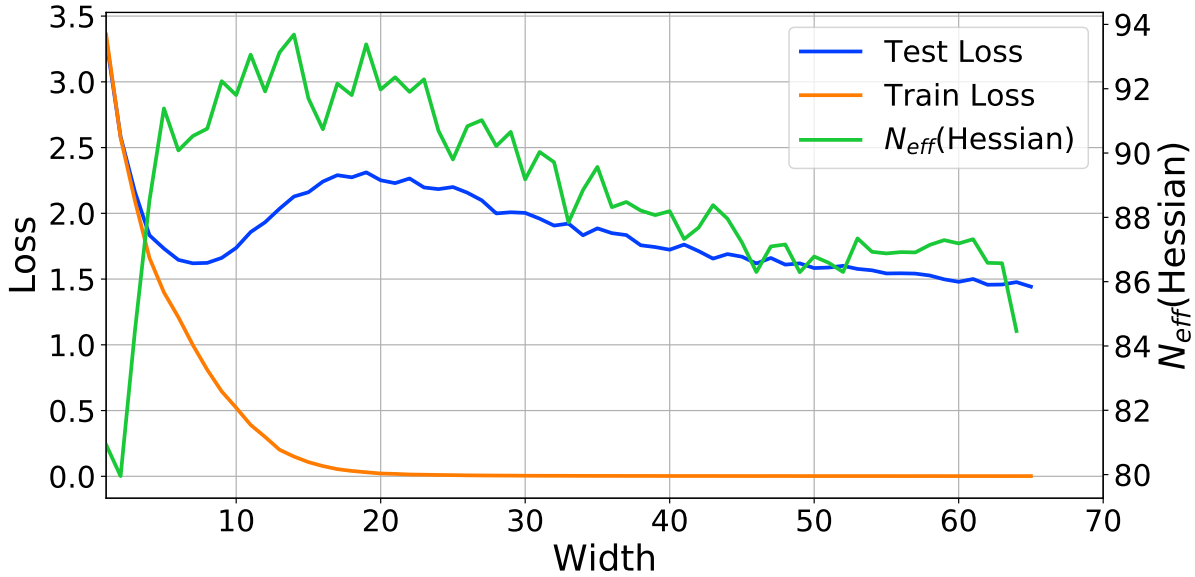
PAC-Bayes provides explicit generalization error bounds for stochastic networks with posterior Q, prior P, training points n, probability $1-\delta$ based on

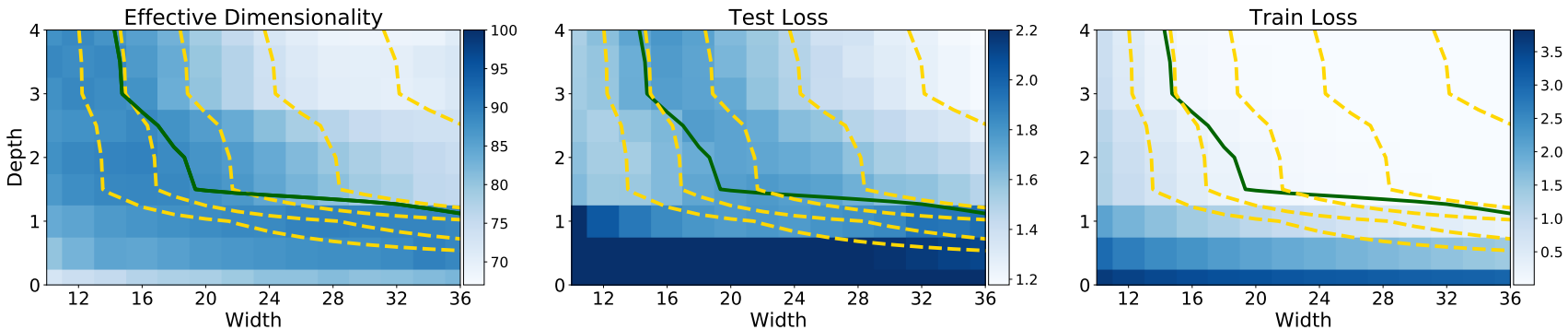
$$\sqrt{\frac{\mathcal{KL}(Q||P) + \log(\frac{n}{\delta})}{2(n-1)}}\,. \tag{6}$$

- Non-vacuous bounds derived from exploiting flatness in Q (e.g., at least 80% generalization accuracy predicted on binary MNIST).
- Very promising framework but tends not to be *prescriptive* about model construction, or informative for understanding *why* a model generalizes.
- Bounds are improved by compact P and a low dimensional parameter space. We suggest a P with large support and many parameters.
- Generalization significantly improved by multimodal Q, but not PAC-Bayes generalization bounds.

Fantastic generalization measures and where to find them. Jiang et. al, 2019.
A primer on PAC-Bayesian learning. Guedj, 2019.
Computing nonvacuous generalization bounds for deep (stochastic) neural networks. Dziugaite & Roy, 2017.
A PAC-Bayesian approach to spectrally-normalized bounds for neural networks. Neyshabur et. al, 2017.

Rethinking Parameter Counting: Effective Dimension



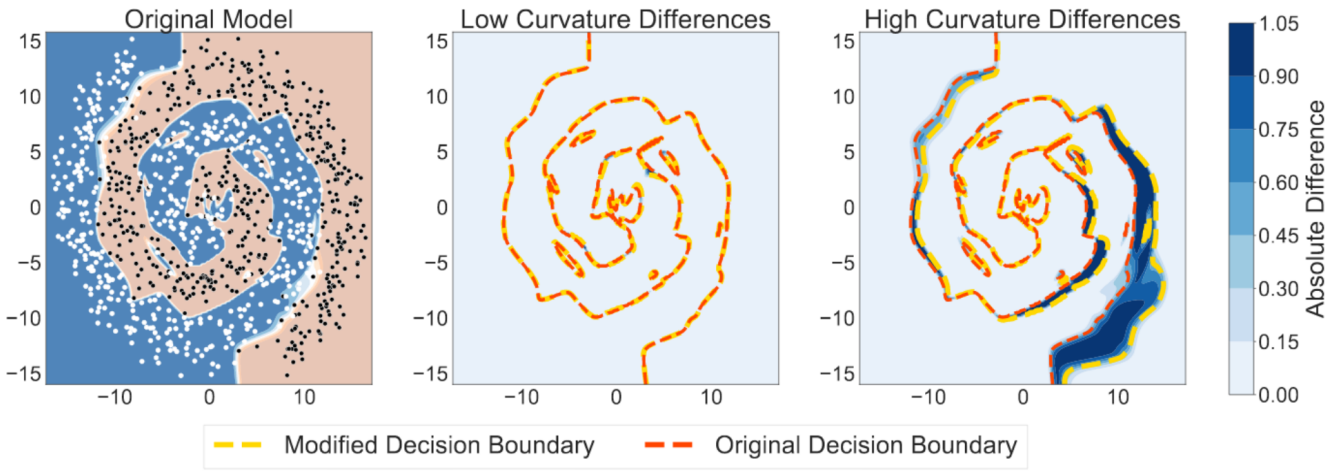


$$N_{\text{eff}}(H) = \sum_i \frac{\lambda_i}{\lambda_i + \alpha}$$

Rethinking Parameter Counting in Deep Models: Effective Dimensionality Revisited.
W. Maddox, G. Benton, A.G. Wilson, 2020.

Properties in Degenerate Directions

Decision boundaries do not change in directions of little posterior contraction, suggesting a mechanism for subspace inference!

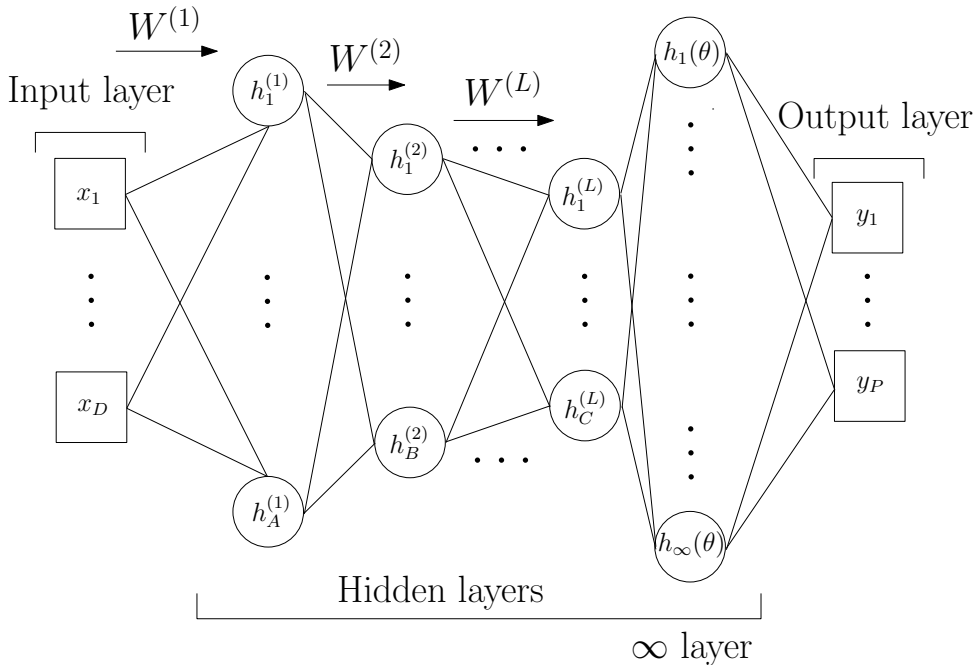


"How can Gaussian processes possibly replace neural networks? Have we thrown the baby out with the bathwater?" (MacKay, 1998)

Introduction to Gaussian processes. MacKay, D. J. In Bishop, C. M. (ed.), Neural Networks and Machine Learning, Chapter 11, pp. 133-165. Springer-Verlag, 1998.

Deep Kernel Learning Review

Deep kernel learning combines the inductive biases of deep learning architectures with the non-parametric flexibility of Gaussian processes.



Base kernel hyperparameters θ and deep network hyperparameters w are jointly trained through the marginal likelihood objective.

Deep Kernel Learning. Wilson, A.G., Hu, Z., Salakhutdinov, R., Xing, E.P. AISTATS, 2016

Face Orientation Extraction

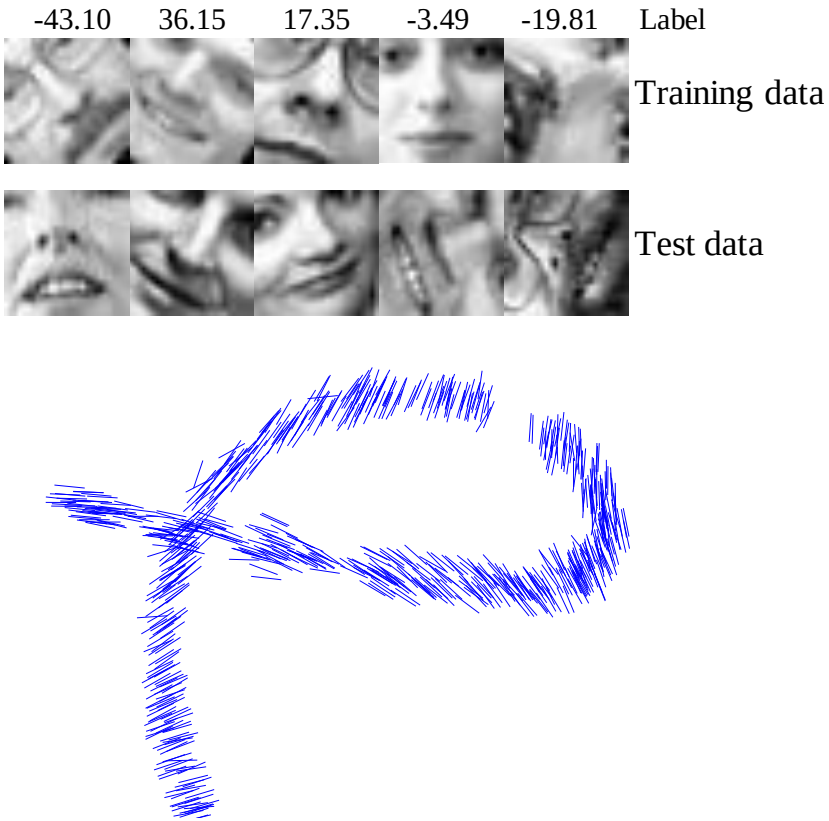


Figure: **Top**: Randomly sampled examples of the training and test data. **Bottom**: The two dimensional outputs of the convolutional network on a set of test cases. Each point is shown using a line segment that has the same orientation as the input face.

Learning Flexible Non-Euclidean Similarity Metrics

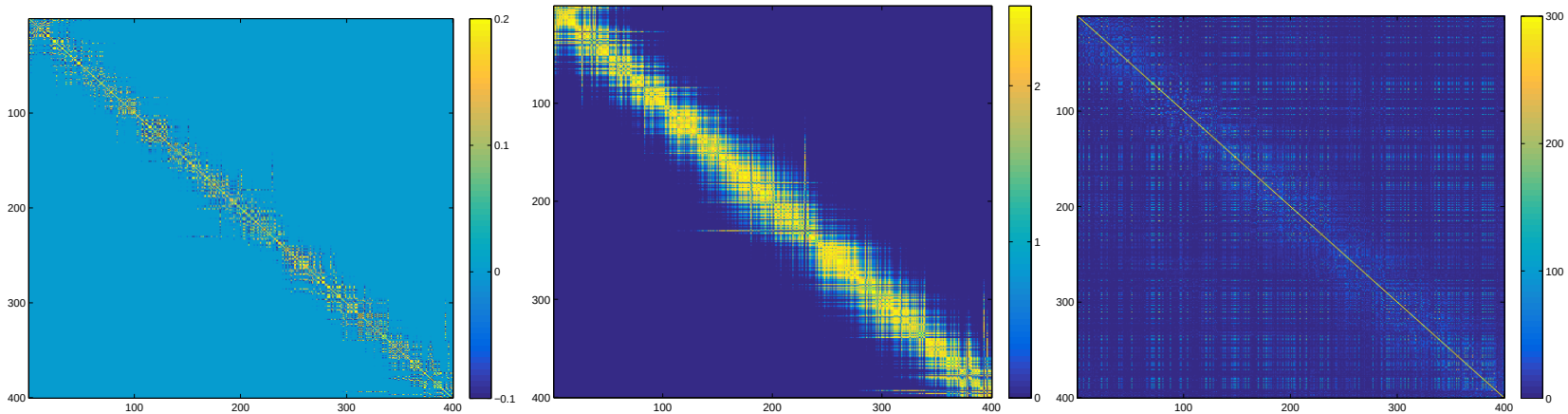


Figure: **Left**: The induced covariance matrix using DKL-SM (spectral mixture) kernel on a set of test cases, where the test samples are ordered according to the *orientations* of the input faces. **Middle**: The respective covariance matrix using DKL-RBF kernel. **Right**: The respective covariance matrix using regular RBF kernel. The models are trained with $n = 12,000$.

Kernels from Infinite Bayesian Neural Networks

- The neural network kernel (Neal, 1996) is famous for triggering research on Gaussian processes in the machine learning community.

Consider a neural network with one hidden layer:

$$f(x) = b + \sum_{i=1}^{J} v_i h(x; \mathbf{u}_i) \,. \tag{7}$$

- b is a bias, v_i are the hidden to output weights, h is any bounded hidden unit transfer function, $\mathbf{u}_i$ are the input to hidden weights, and J is the number of hidden units. Let b and v_i be independent with zero mean and variances σ_b^2 and σ_v^2/J, respectively, and let the $\mathbf{u}_i$ have independent identical distributions.

Collecting all free parameters into the weight vector $\mathbf{w}$,

$$\mathbb{E}_{\mathbf{w}}[f(x)] = 0 \,, \tag{8}$$

$$\operatorname{cov}[f(x), f(x')] = \mathbb{E}_{\mathbf{w}}[f(x)f(x')] = \sigma_b^2 + \frac{1}{J}\sum_{i=1}^{J} \sigma_v^2 \mathbb{E}_{\mathbf{u}}[h_i(x; \mathbf{u}_i)h_i(x'; \mathbf{u}_i)] \,, \tag{9}$$

$$= \sigma_b^2 + \sigma_v^2 \mathbb{E}_{\mathbf{u}}[h(x; \mathbf{u})h(x'; \mathbf{u})] \,. \tag{10}$$

We can show any collection of values $f(x_1), \ldots, f(x_N)$ must have a joint Gaussian distribution using the central limit theorem.

Bayesian Learning for Neural Networks. Neal, R. Springer, 1996.

Neural Network Kernel

$$f(x) = b + \sum_{i=1}^{J} v_i h(x; \mathbf{u}_i) \,. \tag{11}$$

- Let $h(x; \mathbf{u}) = \text{erf}(u_0 + \sum_{j=1}^{P} u_j x_j)$, where $\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$
- Choose $\mathbf{u} \sim \mathcal{N}(0, \Sigma)$

Then we obtain

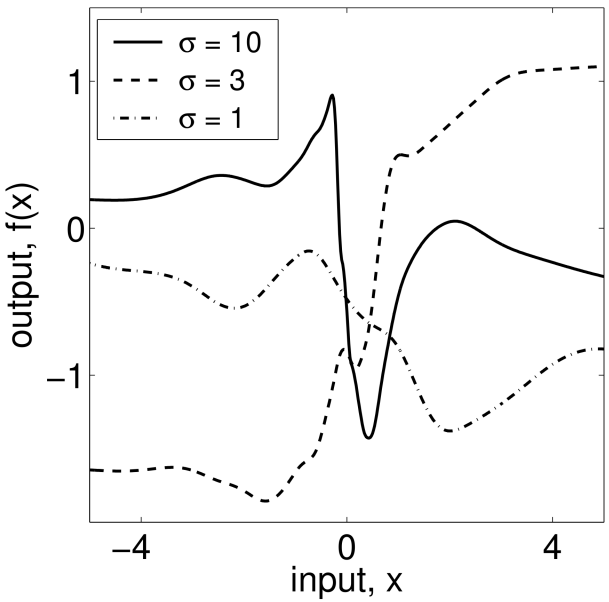
$$k_{\text{NN}}(x, x') = \frac{2}{\pi} \sin\left(\frac{2\tilde{x}^{\text{T}} \Sigma \tilde{x}'}{\sqrt{(1 + 2\tilde{x}^{\text{T}} \Sigma \tilde{x})(1 + 2\tilde{x}'^{\text{T}} \Sigma \tilde{x}')}}\right), \tag{12}$$

where $x \in \mathbb{R}^P$ and $\tilde{x} = (1, x^{\text{T}})^{\text{T}}$.

Neural Network Kernel

$$k_{\mathrm{NN}}(x, x') = \frac{2}{\pi}\sin\left(\frac{2\tilde{x}^{\mathrm{T}}\Sigma\tilde{x}'}{\sqrt{(1+2\tilde{x}^{\mathrm{T}}\Sigma\tilde{x})(1+2\tilde{x}'^{\mathrm{T}}\Sigma\tilde{x}')}}\right) \tag{13}$$

Set $\Sigma = \mathrm{diag}(\sigma_0, \sigma)$. Draws from a GP with a neural network kernel with varying σ:

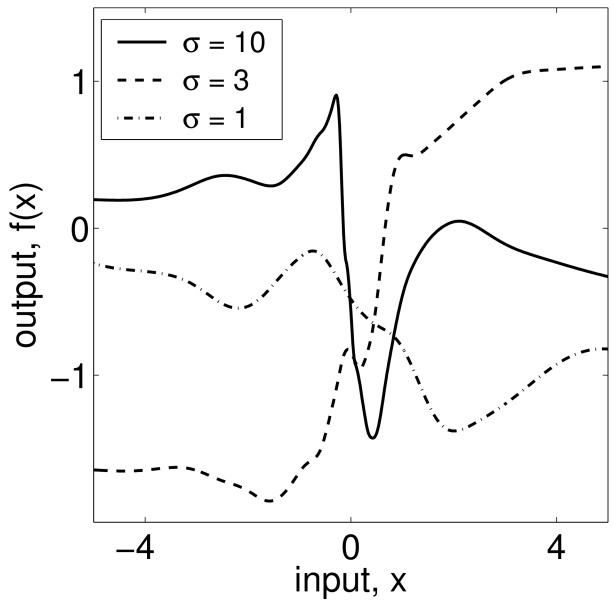


Gaussian processes for Machine Learning. Rasmussen, C.E. and Williams, C.K.I. MIT Press, 2006

Neural Network Kernel

$$k_{\mathrm{NN}}(x, x') = \frac{2}{\pi}\sin\left(\frac{2\tilde{x}^{\mathrm{T}}\Sigma\tilde{x}'}{\sqrt{(1+2\tilde{x}^{\mathrm{T}}\Sigma\tilde{x})(1+2\tilde{x}'^{\mathrm{T}}\Sigma\tilde{x}')}}\right) \tag{14}$$

Set $\Sigma = \mathrm{diag}(\sigma_0, \sigma)$. Draws from a GP with a neural network kernel with varying σ:



Question: Is a GP with this kernel doing representation learning?

Gaussian processes for Machine Learning. Rasmussen, C.E. and Williams, C.K.I. MIT Press, 2006

NN $\rightarrow$ GP Limits and Neural Tangent Kernels

- Several recent works [e.g., 2-9] have extended Radford Neal's limits to multilayer nets and other architectures.
- Closely related work also derives *neural tangent kernels* from infinite neural network limits, with promising results.
- Note that most kernels from infinite neural network limits have a *fixed structure*. On the other hand, standard neural networks essentially *learn* a similarity metric (kernel) for the data. Learning a kernel amounts to *representation learning*. Bridging this gap is interesting future work.

[1] *Bayesian Learning for Neural Networks*. Neal, R. Springer, 1996.
[2] *Deep Convolutional Networks as Shallow Gaussian Processes*. Garriga-Alonso et. al, NeurIPS 2018.
[3] *Gaussian Process Behaviour in Wide Deep Neural Networks*. Matthews et. al, ICLR 2018.
[4] *Deep neural networks as Gaussian processes*. Lee et. al, ICLR 2018.
[5] *Bayesian Deep CNNs with Many Channels are Gaussian Processes*. Novak et. al, ICLR 2019.
[6] *Scaling limits of wide neural networks with weight sharing*. Yang, G. arXiv 2019.
[7] *Neural tangent kernel: convergence and generalization in neural networks*. Jacot et. al, NeurIPS 2018.
[8] *On exact computation with an infinitely wide neural net*. Arora et. al, NeurIPS 2019.
[9] *Harnessing the Power of Infinitely Wide Deep Nets on Small-data Tasks*. Arora et. al, arXiv 2019.

What's next?

- A broader view of deep learning, where we look at deep hierarchical representations, often quite distinct from neural networks.
- Much more Bayesian non-parametric function-space representation learning!
- Challenges will include non-stationarity, high dimensional inputs, scalable high-fidelity approximate inference, and accommodating for misspecification in Bayesian inference procedures.
- Using what we've learned about Gaussian processes as a tool to understand the principles of model construction and a wide variety of model classes.