

Theory: High-probability PAC-Bayesian bound

From the **data distribution** we **sample** the **training set**, which the **model** transforms into a **posterior**, from which we **sample** the **output hypothesis**

$$\mathcal{D} \rightsquigarrow S \xrightarrow{\mathcal{M}} Q \rightsquigarrow h$$

We consider bounds on **generalization error** that hold **with high probability over both sampling steps**

The following **theorem** gives such a bound for the **model** producing the **Bayesian posterior** with 0/1 **likelihood** and any **prior** $P(h)$

For any distribution P on any concept space \mathcal{H} and any realizable distribution \mathcal{D} on a space of instances we have, for $0 < \delta \leq 1$, and $0 < \gamma \leq 1$, that with probability at least $1 - \delta$ over the choice of sample S of m instances, that with probability at least $1 - \gamma$ over the choice of h :

$$\ln(1 - \epsilon(h)) < \frac{\ln \frac{1}{P(C(S))} + \ln m + \ln \frac{1}{\delta} + \ln \frac{1}{\gamma}}{m-1}$$

where

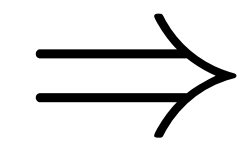
- $C(S)$ is the set of hypotheses in \mathcal{H} consistent with the sample S and $P(C(S)) = \sum_{h \in C(S)} P(h)$
- h is sampled from $Q(h) = \begin{cases} \frac{P(h)}{P(C(S))} & \text{if } h \in C(S) \\ 0 & \text{if } h \notin C(S) \end{cases}$

Proof: Essentially the same as that in (DA McAllister, 1999)

Following (G Valle-Perez et al., 2019), we make the following argument:

If SGD-trained neural networks:

- Reach 0 training error (realizability + trainability)
- Sample the 0-training-error region of parameter space close to uniformly within a bounded domain (unbiasedness in parameter space)



SGD-trained neural networks approximate the above **model**, with **prior** $P(h)$ determined by the parameter-function map upon uniform sampling of inputs.

Implementation

Recent work (J Lee et al., 2017; A Garriga-Alonso et al., 2019; AGG Matthews et al., 2018, R Novak et al., 2019; G Yang, 2019) has shown that **the prior over functions** $P(h)$ upon i.i.d. Gaussian sampling of the weights, **can be approximated by a Gaussian process**, for sufficiently wide neural networks

This is true for almost any type of modern neural network architecture, under appropriate notions of "width" (G Yang, 2019)

We therefore use a Gaussian process to approximately compute $P(C(S))$

The kernel in the Gaussian process for a particular architecture can be written analytically, but is computationally tractable only for some architectures.

For other architectures, we use the **Monte Carlo approximation** proposed in (R Novak et al., 2019):

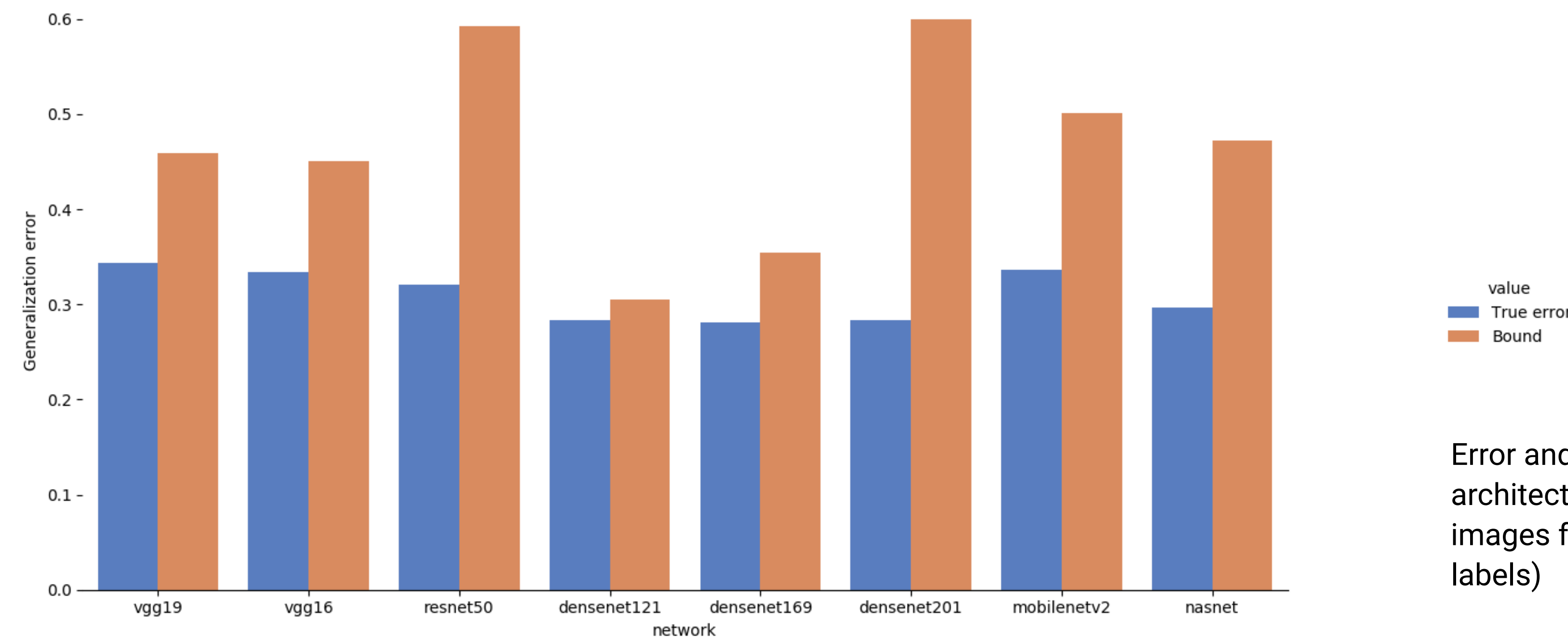
$$\tilde{K}(x, x') = \sum_{i=1}^M h_{\theta_i}(x) h_{\theta_i}(x')$$

where h_{θ_i} is the function computed by the network with parameters θ_i , and we use M samples $\theta_i \sim \mathcal{D}, i = 1, \dots, M$, to compute the empirical covariance of the outputs of the network

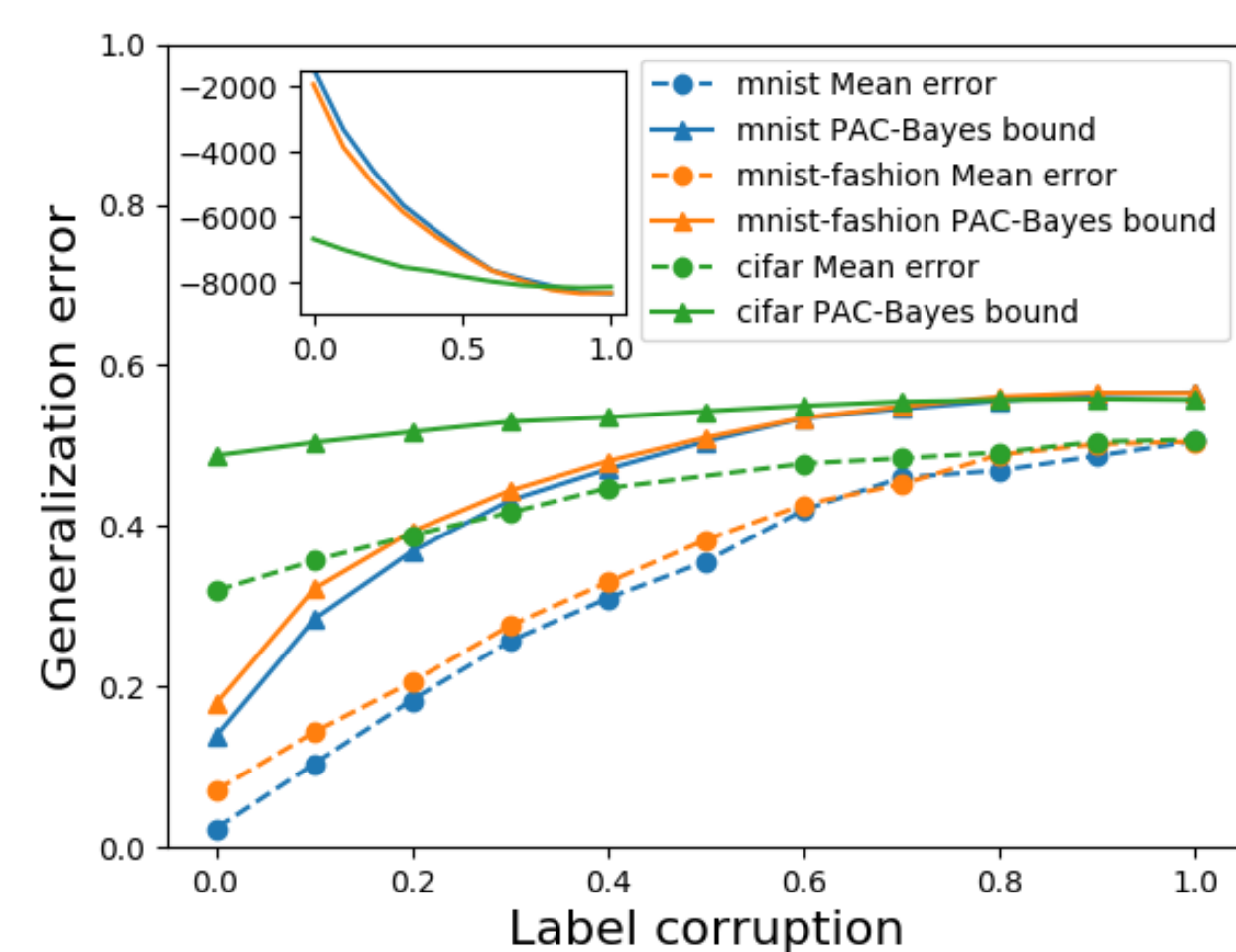
We take M to be some constant times the training set size m to avoid rank deficiency. We use this empirical covariance as an approximation to the true kernel.

Experiments: Tighter bounds on deep learning architectures

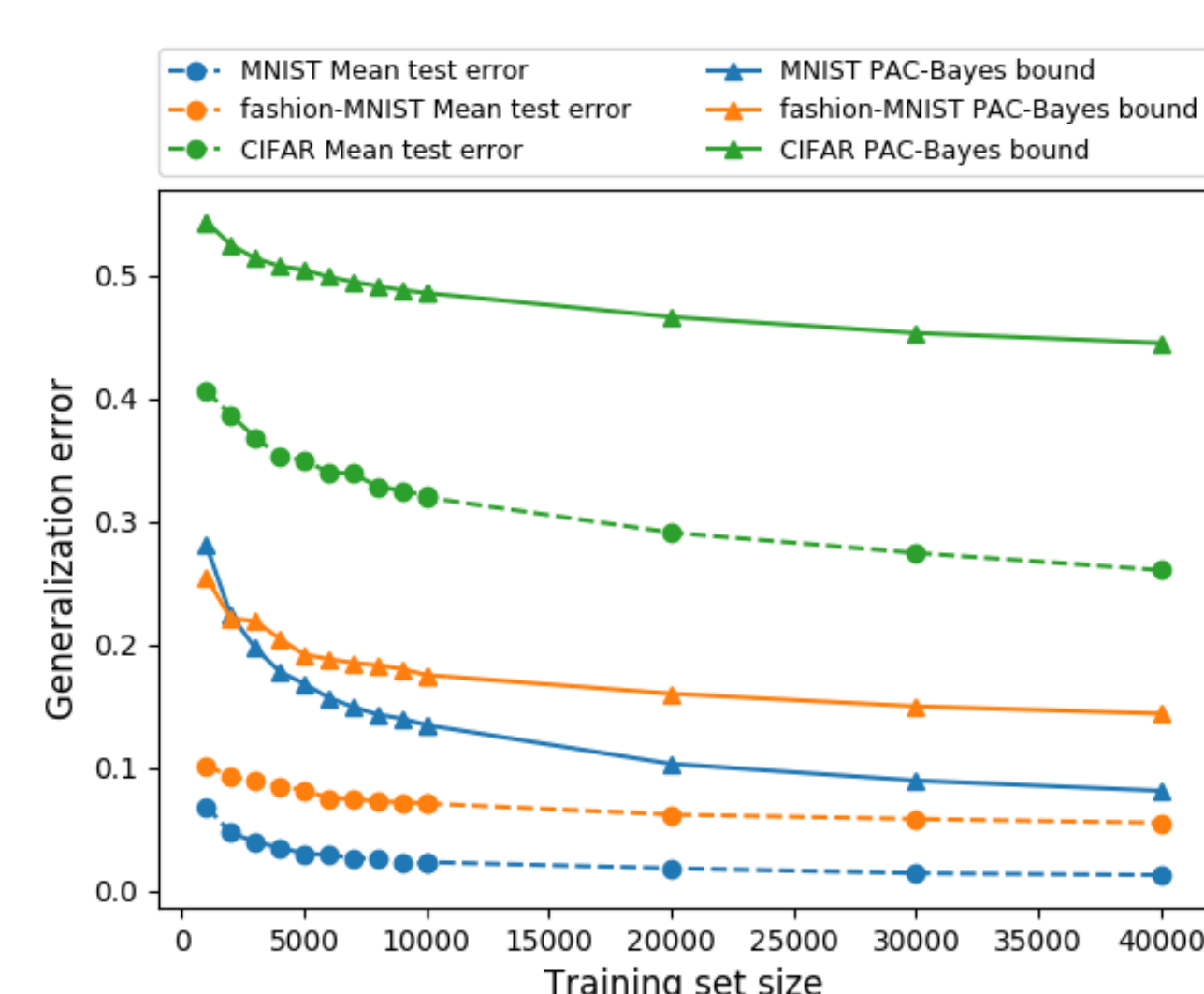
We trained a range of neural network architectures on several standard datasets, and **compared the test error with the PAC-Bayes bound** calculated from the training data.



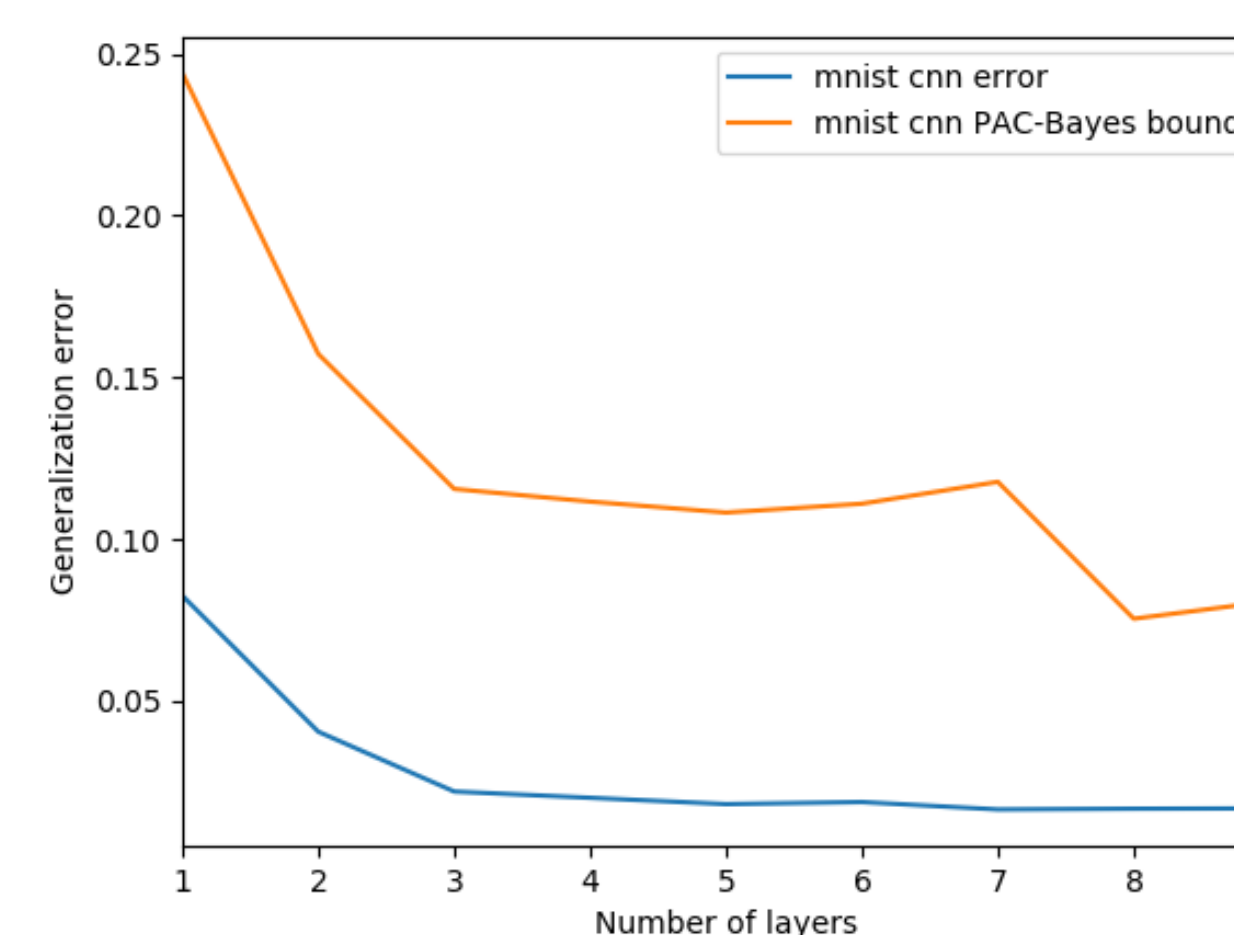
Error and bound, for different architectures trained on a sample of 10k images from CIFAR10 (with binarized labels)



Error and bound, versus label corruption (fraction of target labels which are randomized). Inset shows value of $P(C(S))$. Network is a 4-layer CNN without pooling.



Error and bound, versus training set size m , for a 4-layer CNN without pooling



Error and bound, versus number of layers for a CNN with max pooling, trained on a sample of 10k MNIST images

Limitations

- The bounds only formally apply in the asymptotic limit of infinite width. Making nonasymptotic bounds could prove difficult.
- The bounds depend on the choice of variance of the parameter distribution. This choice seems to have a significant effect only for sufficiently deep neural networks. Understanding how to best choose the variance for different hyperparameter choices is still an open question.
- How well can SGD be approximated by uniform sampling in the zero-error region is also an open question.
- The calculation of the marginal likelihood is approximate, using techniques which typically don't have rigorous guarantees (expectation propagation, MCMC). In this work we used expectation-propagation, but an MCMC approach is probably more accurate (at the expense of computation time).
- It isn't clear how tight PAC-Bayes itself is, as no matching lower bounds are available.
- As discussed by (Kääriäinen et al., 2005), the practical value of PAC-Bayesian versus test-set generalization error bounds may be limited. This is related to the above question of the tightness of the bounds.

Refs:

- Valle-Perez et al., 2019.** Deep learning generalizes because the parameter-function map is biased towards simple functions. Published in ICLR 2019
- J Lee et al., 2017.** Deep neural networks as gaussian processes. arXiv preprint arXiv:1711.00165, 2017.
- A Garriga-Alonso et al., 2019.** Deep convolutional networks as shallow Gaussian processes. Published in ICLR 2019
- R Novak et al., 2019.** Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes. Published in ICLR 2019
- AGG Mathews et al., 2018.** Gaussian Process Behaviour in Wide Deep Neural Networks. Published in ICLR 2018
- G Yang, 2019.** Scaling Limits of Wide Neural Networks with Weight Sharing: Gaussian Process Behavior, Gradient Independence, and Neural Tangent Kernel Derivation. arXiv preprint arXiv:1902.04760, 2019
- M Kääriäinen et al., 2005.** A comparison of tight generalization error bounds. Published in ICML 2005
- DA McAllister, 1999.** Some pac-bayesian theorems. Machine Learning (1999) 37: 355