Conditionally Gaussian PAC-Bayes

E. Clerico, G. Deligiannidis, A. Doucet

University of Oxford, Department of Statistics

AISTATS 2022

Stochastic network

Stochastic neural classifiers: the parameters are random variables.

$$\text{PRIOR} \quad \xrightarrow[\text{(dataset } S)]{\text{training}} \quad \text{POSTERIOR}$$

PAC-Bayesian bounds

$$\begin{array}{l}\text{Expected}\\ \text{true error}\end{array} \;\leq\; \mathcal{B}\left(\begin{array}{l}\text{Expected}\\ \text{empirical}\\ \text{error}\end{array}, \mathrm{KL}(\text{Posterior}, \text{Prior})\right),$$

with high probability on the random draw of the sample S.

Standard PAC-Bayesian training

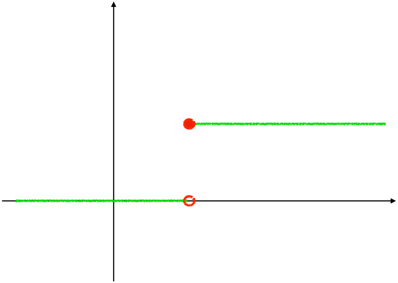
Under Gaussian prior and posterior:

- Sample the parameters;
- Evaluate a realisation of $\mathcal{B}$;
- GD step.

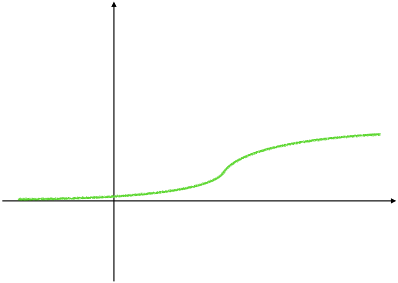
Null gradient issue

$\mathcal{B}$'s estimate has a null gradient $\implies$ surrogate loss

Misclassification loss: $\nabla \ell = 0$



Surrogate loss: $\nabla \ell \neq 0$



But... Mismatch between the training objective and $\mathcal{B}$!

The gradient of the expected empirical error is non-null!

We need a different estimate for $\mathcal{B}$

Main ideas:

- If the output is Gaussian, then we can compute the expected empirical error's gradient;
- The output is Gaussian when conditioned on the last hidden layer's output.

Cond-Gauss algorithm

- Sample the random parameters of all the layers, but the last one;
- Compute the gradient exploiting the conditional Gaussianity;
- Perform the GD step.

Empirical results

The bounds that we found with the Cond-Gauss algorithm were tighter than those obtained via other state-of-the-art PAC-Bayesian methods, on both MNIST and CIFAR-10.

dataset	architecture	prior	C-G	P-O
MNIST	4 layers	data-free	.1348	.2165
MNIST	4 layers	50%	.0144	.0155
CIFAR10	9 layers	50%	.2066	.2901
CIFAR10	15 layers	50%	.1855	.1954
CIFAR10	15 layers	70%	.1595	.1667