Wide stochastic networks: Gaussian limit and PAC-Bayesian training

E. Clerico, G. Deligiannidis, A. Doucet

University of Oxford, Department of Statistics

ALT 2023

Overparameterised model

Overparameterised regime:

- Many more parameters than training datapoints
- Typical of modern NNs
- Generalise "better than expected"
- Extremely complex mathematical problem
- Limit asymptotic regimes sometimes more "tractable" (*e.g.* infinite width)

Infinite-width limit

For a feedworward network we let tend the number of nodes of each layer to infinity: **infinite width limit**

Under suitably scaled iid initialisation:

- Gaussian behaviour at initialisation [Neal, 1995]
- NTK regime during training [Jacot et al., 2018]

Our paper: Gaussian asymptotics for *wide shallow stochastic NN*

Stochastic NN

The trainable parameters are **random variables**

$$F(x) = W^1 \phi(W^0 x)$$

W^0 is a $n \times p$ matrix: $W^0_{ij} = \frac{1}{\sqrt{n}}(\mathfrak{s}^0_{jk}\zeta^0_{jk} + \mathfrak{m}^0_{jk})$
W^1 is a $q \times n$ matrix: $W^1_{jk} = \frac{1}{\sqrt{p}}(\mathfrak{s}^1_{ij}\zeta^1_{ij} + \mathfrak{m}^1_{ij})$

All the ζ's are iid $\sim \mathcal{N}(0,1)$, $\mathfrak{m}$ and $\mathfrak{s}$ are **deterministic**

Infinite width limit

Infinite width limit: $n \to \infty$

Informally: $\forall x, \quad F(x) \to \mathcal{N}(M(x), Q(x))$

Gaussian behaviour:

- At **initialisation**
- Throughout **lazy training**

Nota bene

Two sources of randomness

- Initialisation $\hat{\mathbb{P}}$,
- Intrinsic stochasticity $\mathbb{P}$

In the standard setting infinite witdh limit is Gaussian wrt $\hat{\mathbb{P}}$. It is **deterministic** conditioned on $\hat{\mathbb{P}}$.

Here, we will condition on the initialisation drawn from $\hat{\mathbb{P}}$, and find a Gaussian limit wrt $\mathbb{P}$.

Hidden layer

$$Y_j^0(x) = \sum_{k=1}^p W_{jk}^0 x_k = \frac{1}{\sqrt{p}} \sum_{k=1}^p \mathfrak{s}_{jk}^0 \zeta_{jk}^0 x_k + \frac{1}{\sqrt{p}} \sum_{k=1}^p \mathfrak{m}_{jk}^0 x_k$$

Y_j^0 is the sum of finitely many Gaussians...

$$Y^0(x) \sim \mathcal{N}(M^0(x), Q^0(x))$$

$$M_j^0(x) = \frac{1}{\sqrt{p}} \sum_{k=1}^p \mathfrak{m}_{jk}^0 x_k$$

$$Q_{jj'}^0(x) = \delta_{jj'} \frac{1}{p} \sum_{k=1}^p (\mathfrak{s}_{jk}^0 x_k)^2$$

Output

$$F_i(x) = \sum_{j=1}^n W_{ij}^1 \Phi_j^0(x) = \frac{1}{\sqrt{n}} \sum_{j=1}^n \mathfrak{s}_{1j}^1 \zeta_{ij}^1 \Phi_j^0(x) + \frac{1}{\sqrt{n}} \sum_{j=1}^n \mathfrak{m}_{ij}^1 \Phi_j^0(x)$$

with $\Phi_j^0(x) = \phi(Y_j^0(x))$

This is a sum of **independent** RVs, but not iid!
Need a *Lyapunov-like* CLT

$$M_i(x) = \frac{1}{\sqrt{n}} \sum_{j=1}^n \mathfrak{m}_{ij}^1 \mathbb{E}[\Phi_j^0(x)]$$
$$Q_{ii'}(x) = \delta_{ii'} \frac{1}{n} \sum_{j=1}^n (\mathfrak{s}_{ij}^1)^2 \mathbb{E}[\Phi_j^0(x)^2] + \frac{1}{n} \sum_{j=1}^n \mathfrak{m}_{ij}^1 \mathfrak{m}_{i'j}^1 \mathbb{V}[\Phi_j^0(x)]$$

CLT

Proposition (CLT, Benktus (2005))

x and n fixed. $Z(x) \sim \mathcal{N}(M(x), Q(x))$ and $\mathcal{C}$ the class of measurable convex subsets of $\mathbb{R}^q$. Then

$$\sup_{C \in \mathcal{C}} |\mathbb{P}(F(x) \in C) - \mathbb{P}(Z(x) \in C)| \leq 4q^{1/4} \frac{B(\mathfrak{m}, \mathfrak{s})}{\sqrt{n}} .$$

In particular, if $B(\mathfrak{m}, \mathfrak{s}) = O(1)$ for $n \to \infty$, then $F(x) \to Z(x)$, in distribution.

Initialisation

$$\mathfrak{m}^0_{jk} \sim \mathcal{N}(0,1)\,; \qquad \mathfrak{m}^1_{ij} \sim \mathcal{N}(0,1)$$
$$\mathfrak{s}^0_{jk} = 1\,; \qquad \mathfrak{s}^1_{ij} = 1$$

Proposition (Initialisation)

Consider a sequence of networks of increasing width initialised as above, and whose activation function ϕ is Lipshitz continuous. For any fixed input $x \neq 0$, we have $\frac{B(\mathfrak{m},\mathfrak{s})}{\sqrt{n}} \to 0$, as $n \to \infty$, in probability with respect to the random initialisation $\hat{\mathbb{P}}$. More precisely, $B(\mathfrak{m},\mathfrak{s}) = O(1)$ wrt $\hat{\mathbb{P}}$, as $n \to \infty$. In particular, at the initialisation the network tends to a Gaussian limit, in distribution wrt the intrinsic stochasticity $\mathbb{P}$ and in probability wrt $\hat{\mathbb{P}}$.

Proof's sketch.

Hyper-parameters iid at init. By CLT B upperbounded by a finite limit as $n \to \infty$. □

Lazy training

Proposition (Lazy training)

Fix a constant $J > 0$ independent of n, and assume that ϕ is Lipschitz. Initial configuration $(\widetilde{\mathfrak{m}}, \widetilde{\mathfrak{s}})$ drawn according to $\hat{\mathbb{P}}$. $\mathcal{B}_J$ the ball

$$\mathcal{B}_J = \left\{(\mathfrak{m}, \mathfrak{s}) \ : \quad \|\mathfrak{m} - \widetilde{\mathfrak{m}}\|_{F,2}^2 + \|\mathfrak{s} - \widetilde{\mathfrak{s}}\|_{F,2}^2 \leq J^2\right\} .$$

For any fixed input $x \neq 0$ we have $B(\mathfrak{m}, \mathfrak{s}) = O(1)$ as $n \to \infty$, uniformly on $\mathcal{B}_J$, in probability with respect to the random initialisation $\hat{\mathbb{P}}$.

Proof's sketch.

The proof is technical, but the idea is simple and consists in showing that B undergoes a change of order $O(1)$ during the training, under the lazy training assumption $(\mathfrak{m}, \mathfrak{s}) \in \mathcal{B}_J$. Since we know that B is of order $O(1)$ at the initialisation, we can conclude. □

Infinite width: summary

- $F(x) \sim \mathcal{N}(M(x), Q(x))$
- $M(x)$ and $Q(x)$ computable wrt $\mathfrak{m}$, $\mathfrak{s}$
- Holds for initialisation and lazy training

PAC-Bayes

Framework for generalisation bounds for stochastic networks.

- π, ρ prior and posterior laws on the random parameters
- π is **data-agnostic**
- ρ is **data-dependent**

Idea: if the algorithm does not leak too much information from the data then it will generalise well. Amount of *leaked information* here is represented by how far ρ is from π.

Simple example: for a bounded loss function $\ell \subseteq [0, 1]$

$$\mathbb{E}_\rho[\mathscr{L}_X] \leq \mathbb{E}_\rho[\mathscr{L}_S] + \frac{1}{\sqrt{m}} \left(\mathrm{KL}(\rho \| \pi) + \log \frac{1}{\delta} + \frac{1}{8} \right)$$

with probability at least $1 - \delta$ on S.

PAC-Bayesian training

We can use a PAC-Bayesian bound as training objective for a **stochastic network**

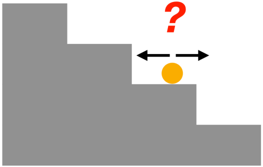
- Non-vacuous bounds for overparameterised networks
- Requires specific stochastic architectures
- Need to evaluate $\mathbb{E}_\rho[\mathscr{L}_S]$ and $\mathrm{KL}(\rho\|\pi)$ and their gradients

[Dziugaite and Roy, 2017; Pérez-Ortiz et al., 2021]

Common issues for PAC-Bayesian training

- $\mathrm{KL}(\rho \| \pi)$ has a closed form for Gaussian parameters, but $\mathbb{E}_\rho[\mathscr{L}_S]$ is not know for a general ρ.
- Usually output's law is unknown and $\mathbb{E}_\rho[\mathscr{L}_S]$ needs MC sampling.
- Estimating $\nabla \mathbb{E}_\rho[\mathscr{L}_S]$ might require **surrogate loss** .

$\implies$ There is a **mismatch** between the bound and the objective.

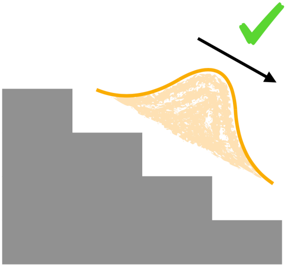


Gaussian PAC-Bayes

- For infinite width limit the output is Gaussian at initialisation.
- PAC-Bayesian training is lazy when “prior=init”:

$$\|\Delta\mathfrak{m}\|_{F,2}^2 + \|\Delta\mathfrak{s}\|_{F,2}^2 \leq 2\mathrm{KL}(\rho\|\pi)$$

If output’s law is known $\implies$ *informative* gradient with 01-loss.



Training idea

- We can train a shallow wide stochastic network by *pretending* that it has a Gaussian output and optimise a PAC-Bayes bound.
- Actually it will only be approximately Gaussian, so in order to obtain an exact bound at the end we will need to take this into account rigorously.

Gradient wrt M and Q

Binary classification problem: prediction $\mathrm{argmax}_i\, F_i(x)$.

$$\mathbb{E}[\ell(\hat{f}(x), 1)] = \mathbb{P}_{\zeta\sim\mathcal{N}(0,1)}\left(\zeta > \frac{M_1(x) - M_2(x)}{\sqrt{Q_{11}(x) + Q_{22}(x) - 2Q_{12}(x)}}\right)$$

This is a differentiable function of M and Q, whose gradient can be computed explicitly, as $\mathbb{P}(\zeta > u) = \frac{1}{2}(1 - \mathrm{erf}(u/\sqrt{2}))$.

Gradient wrt $\mathfrak{m}$ and $\mathfrak{s}$

- Recall that M and Q contain terms in the form $\mathbb{E}[\Phi_j(x)]$ and $\mathbb{E}[\Phi_j(x)^2]$, with $\Phi_j(x) = \phi(Y_j^0(x))$.
- We have $Y_j^0(x) \sim \mathcal{N}(M_j^0(x), \sqrt{Q_{jj}^0(x)})$.
- For simple enough ϕ, $\mathbb{E}[\phi(a\zeta + b)]$ can be computed.

$$\implies \nabla_{\mathfrak{m},\mathfrak{s}} \text{ can be computed analytically...}$$

Experimental results

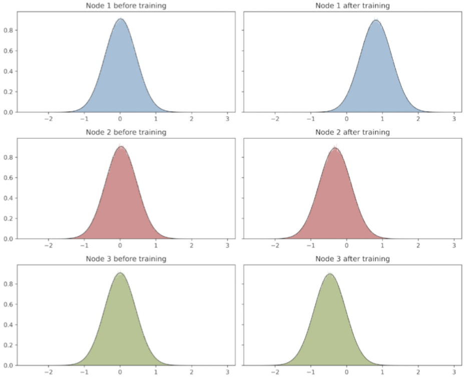


Table 1: Binary MNIST

Method	Bound	Test error	G Bound	G Loss	Penalty
invkl	**.1773**	$.0694_{\pm.0040}$	.1741	.0676	.0492
McAll	.1978	$\mathbf{.0456}_{\pm.0025}$	.1947	.0428	.1006
lbd	.1856	$.0543_{\pm.0030}$	.1825	.0520	.0752
quad	.1855	$.0533_{\pm.0030}$	.1823	.0515	.0757

Table 2: MNIST

Method	Bound	Test Error	G Bound	G Loss	Penalty
invkl	**.2807**	$\mathbf{.1083}_{\pm.0039}$	.2773	.1114	.0821
McAll	.4158	$.3189_{\pm.0097}$	.4120	.3265	.0155
lbd	.3736	$.2639_{\pm.0085}$	.3699	.2717	.0216
quad	.3735	$.2637_{\pm.0083}$	.3698	.2716	.0217

Conclusion

- $F(x) \to \mathcal{N}(M(x), Q(x))$ at init and under lazy training
- Application: PAC-Bayesian training
- Issue: limit cannot be easily extended to multilayer networks
- Gaussian PAC-Bayesian training method inspired conditionally Gaussian method for multilayer architectures [Clerico et al., 2022]
- M and Q can be seen as output of deterministic neural network with complex activations

Thank you :)