PAC-Bayesian Learning and Neural Networks

The Binary Activated Case

Pascal Germain[1,2]

joint work with Gaël Letarte[3], Benjamin Guedj[1,2,4], François Laviolette[3]

[1] **Inria Lille - Nord Europe (équipe-projet MODAL)**
[2] **Laboratoire Paul Painlevé (équipe Probabilités et Statistique)**
[3] Université Laval (équipe GRAAL) [4] University College London (CSML group)

51es Journées de Statistique de la SFdS (Nancy)

APRIORI
A PAC-Bayesian representation learning perspective

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR

June 6, 2019

Plan

Plan

Setting

Assumption : Learning samples are generated *iid* by a data-distribution D.

$$S = \{ (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n) \} \sim D^n$$

Objective

Given a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, find a predictor $f \in \mathcal{F}$ that minimizes the **generalization loss** on D :

$$\mathcal{L}_D(f) := \mathop{\mathbf{E}}_{(x,y)\sim D} \ell(f(x), y)$$

Challenge

The learning algorithm has *only* access to the **empirical loss** on S

$$\widehat{\mathcal{L}}_S(f) := \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

PAC-Bayesian Theory

Pioneered by
Shawe-Taylor et Williamson (1997), McAllester (1999) et Catoni (2003),
the PAC-Bayesian theory give **PAC** generalization guarantees to "**Bayesian** like" algorithms.

PAC guarantees (Probably Approximately Correct)

With probability at least "$1-\delta$", the loss of predictor f is less than "ε"

$$\Pr_{S\sim D^n}\Big(\mathcal{L}_D(f) \leq \varepsilon(\widehat{\mathcal{L}}_S(f), n, \delta, \ldots)\Big) \geq 1-\delta$$

Bayesian flavor

Given :

- A **prior** distribution P on $\mathcal{F}$.
- A **posterior** distribution Q on $\mathcal{F}$.

$$\Pr_{S\sim D^n}\Big(\mathop{\mathbf{E}}_{f\sim Q}\mathcal{L}_D(f) \leq \varepsilon(\mathop{\mathbf{E}}_{f\sim Q}\widehat{\mathcal{L}}_S(f), n, \delta, P, \ldots)\Big) \geq 1-\delta$$

A *Classical* PAC-Bayesian Theorem

PAC-Bayesian theorem (adapted from McAllester 1999 ; McAllester 2003)

For any distribution P on $\mathcal{F}$, for any $\delta \in (0,1]$, we have,

$$\Pr_{S\sim D^n}\left(\forall\, Q \text{ on } \mathcal{F} : \mathop{\mathbf{E}}_{f\sim Q} \mathcal{L}_D(f) \leq \underbrace{\mathop{\mathbf{E}}_{f\sim Q} \widehat{\mathcal{L}}_S(f)}_{\text{empirical loss}} + \underbrace{\sqrt{\frac{1}{2n}\left[\mathrm{KL}(Q\|P) + \ln\frac{2\sqrt{n}}{\delta}\right]}}_{\text{complexity term}}\right) \geq 1-\delta\,,$$

where $\mathrm{KL}(Q\|P) = \mathop{\mathbf{E}}_{f\sim Q} \ln\frac{Q(f)}{P(f)}$ is the **Kullback-Leibler divergence**.

Training bound

- Gives generalization guarantees **not based on testing sample**.

Valid *for all* posterior Q on $\mathcal{F}$

- Inspiration for conceiving **new learning algorithms**.

Plan

Standard Neural Networks (Multilayer perceptrons, or MLP)

Classification setting :

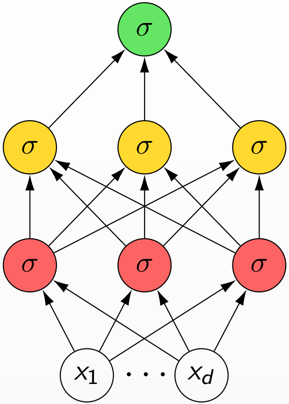
- $\mathbf{x} \in \mathbb{R}^d$
- $y \in \{-1, 1\}$

Architecture :

- *L fully connected* layers
- d_k denotes the number of neurons of the k^{th} layer
- $\sigma : \mathbb{R} \to \mathbb{R}$ is the *activation function*

Parameters :

- $\mathbf{W}_k \in \mathbb{R}^{d_k \times d_{k-1}}$ denotes the weight matrices.
- $\theta = \mathrm{vec}\left(\{\mathbf{W}_k\}_{k=1}^L\right) \in \mathbb{R}^D$



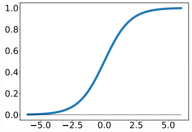
Prediction

$$f_\theta(\mathbf{x}) = \sigma\Big(\mathbf{w}_L \sigma\Big(\mathbf{W}_{L-1} \sigma\Big(\ldots \sigma\Big(\mathbf{W}_1 \mathbf{x}\Big)\Big)\Big)\Big)\,.$$

First PAC-Bayesian bounds for Stochastic Neural Networks

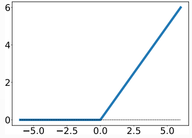
"(Not) Bounding the True Error" (LANGFORD et CARUANA 2001)

- Shallow networks ($L = 2$)
- Sigmoid activation functions



"Computing Nonvacuous Generalization Bounds for Deep **(Stochastic)** Neural Networks" (DZIUGAITE et ROY 2017)

- Deep networks ($L > 2$)
- ReLU activation functions



Idea : Bound the expected loss of the network under a Gaussian perturbation of the weights

Empirical loss : $\mathop{\mathbf{E}}_{\theta' \sim \mathcal{N}(\theta, \Sigma)} \widehat{\mathcal{L}}_S(f_{\theta'})$ $\longrightarrow$ estimated by sampling

Complexity term : $\mathrm{KL}(\mathcal{N}(\theta, \Sigma) \| \mathcal{N}(\theta_p, \Sigma_p))$ $\longrightarrow$ closed form

Plan

Binary Activated Neural Networks

Classification setting :

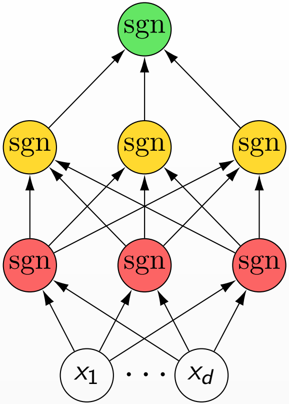
- $\mathbf{x} \in \mathbb{R}^d$
- $y \in \{-1, 1\}$

Architecture :

- L *fully connected* layers
- d_k denotes the number of neurons of the k^{th} layer
- $\mathrm{sgn}(a) = 1$ if $a > 0$ and $\mathrm{sgn}(a) = -1$ otherwise

Parameters :

- $\mathbf{W}_k \in \mathbb{R}^{d_k \times d_{k-1}}$ denotes the weight matrices.
- $\theta = \mathrm{vec}\big(\{\mathbf{W}_k\}_{k=1}^L\big) \in \mathbb{R}^D$



Prediction

$$f_\theta(\mathbf{x}) = \mathrm{sgn}\big(\mathbf{w}_L \mathrm{sgn}\big(\mathbf{W}_{L-1} \mathrm{sgn}\big(\ldots \mathrm{sgn}\big(\mathbf{W}_1 \mathbf{x}\big)\big)\big)\big)\,,$$

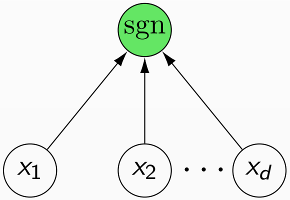
Plan

1. PAC-Bayesian Learning
2. *Standard* Neural Networks
3. *Binary Activated* Neural Networks
   - One Layer (Linear predictor)
   - Two Layers (shallow)
   - More Layers (deep)
4. Empirical results

One Layer

"PAC-Bayesian Learning of Linear Classifiers" (Germain et al. 2009)

$f_{\mathbf{w}}(\mathbf{x}) := \mathrm{sgn}(\mathbf{w} \cdot \mathbf{x})$, with $\mathbf{w} \in \mathbb{R}^d$.



One Layer

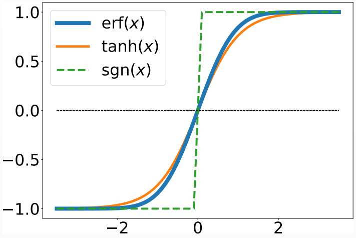
"PAC-Bayesian Learning of Linear Classifiers" (Germain et al. 2009)

$$f_{\mathbf{w}}(\mathbf{x}) := \mathrm{sgn}(\mathbf{w}\cdot\mathbf{x}), \text{ with } \mathbf{w} \in \mathbb{R}^d.$$

PAC-Bayes analysis :

- Space of all linear classifiers $\mathcal{F}_d := \{f_{\mathbf{v}} | \mathbf{v} \in \mathbb{R}^d\}$
- Gaussian posterior $Q_{\mathbf{w}} := \mathcal{N}(\mathbf{w}, I_d)$ over $\mathcal{F}_d$
- Gaussian prior $P_{\mathbf{w}_0} := \mathcal{N}(\mathbf{w}_0, I_d)$ over $\mathcal{F}_d$
- Predictor

$$F_{\mathbf{w}}(\mathbf{x}) := \mathop{\mathbf{E}}_{\mathbf{v}\sim Q_{\mathbf{w}}} f_{\mathbf{v}}(\mathbf{x}) = \mathrm{erf}\left(\frac{\mathbf{w}\cdot\mathbf{x}}{\sqrt{d}\|\mathbf{x}\|}\right)$$



Bound minimization — under the linear loss $\ell(y, y') := \frac{1}{2}(1 - yy')$

$$C\, n\, \widehat{\mathcal{L}}_S(F_{\mathbf{w}}) + \mathrm{KL}(Q_{\mathbf{w}} \| P_{\mathbf{w}_0}) \;=\; C\, \frac{1}{2}\sum_{i=1}^{n} \mathrm{erf}\left(-y_i\, \frac{\mathbf{w}\cdot\mathbf{x}_i}{\sqrt{d}\|\mathbf{x}_i\|}\right) + \frac{1}{2}\|\mathbf{w} - \mathbf{w}_0\|^2 .$$

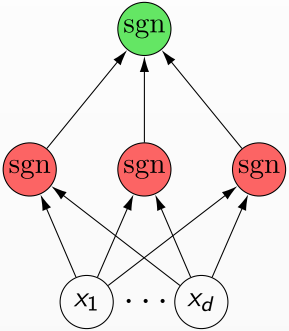
Plan

1. PAC-Bayesian Learning
2. *Standard* Neural Networks
3. *Binary Activated* Neural Networks
   - One Layer (Linear predictor)
   - Two Layers (shallow)
   - More Layers (deep)
4. Empirical results

Posterior $Q_\theta = \mathcal{N}(\theta, I_D)$, over the family of all networks $\mathcal{F}_D = \{f_{\tilde\theta} \,|\, \tilde\theta \in \mathbb{R}^D\}$, where

$$f_\theta(\mathbf{x}) = \operatorname{sgn}\big(\mathbf{w}_2 \cdot \operatorname{sgn}(\mathbf{W}_1\mathbf{x})\big)\,.$$

$$\begin{aligned}
F_\theta(\mathbf{x}) &= \mathop{\mathbf{E}}_{\tilde\theta\sim Q_\theta} f_{\tilde\theta}(\mathbf{x})\\
&= \int_{\mathbb{R}^{d_1\times d_0}} Q_1(\mathbf{V}_1) \int_{\mathbb{R}^{d_1}} Q_2(\mathbf{v}_2)\operatorname{sgn}(\mathbf{v}_2 \cdot \operatorname{sgn}(\mathbf{V}_1\mathbf{x}))d\mathbf{v}_2 d\mathbf{V}_1\\
&= \int_{\mathbb{R}^{d_1\times d_0}} Q_1(\mathbf{V}_1) \operatorname{erf}\left(\frac{\mathbf{w}_2\cdot\operatorname{sgn}(\mathbf{V}_1\mathbf{x})}{\sqrt{2}\|\operatorname{sgn}(\mathbf{V}_1\mathbf{x})\|}\right) d\mathbf{V}_1\\
&= \sum_{\mathbf{s}\in\{-1,1\}^{d_1}} \operatorname{erf}\left(\frac{\mathbf{w}_2\cdot\mathbf{s}}{\sqrt{2d_1}}\right) \int_{\mathbb{R}^{d_1\times d_0}} \mathbb{1}[\mathbf{s} = \operatorname{sgn}(\mathbf{V}_1\mathbf{x})]\, Q_1(\mathbf{V}_1)\, d\mathbf{V}_1\\
&= \sum_{\mathbf{s}\in\{-1,1\}^{d_1}} \underbrace{\operatorname{erf}\left(\frac{\mathbf{w}_2\cdot\mathbf{s}}{\sqrt{2d_1}}\right)}_{F_{\mathbf{w}_2}(\mathbf{s})} \underbrace{\prod_{i=1}^{d_1}\left[\frac{1}{2} + \frac{s_i}{2}\operatorname{erf}\left(\frac{\mathbf{w}_1^i\cdot\mathbf{x}}{\sqrt{2}\,\|\mathbf{x}\|}\right)\right]}_{\Pr(\mathbf{s}|\mathbf{x},\mathbf{W}_1)}.
\end{aligned}$$



PAC-Bayesian Ingredients

Empirical loss

$$\widehat{\mathcal{L}}_S(F_\theta) = \mathop{\mathbf{E}}_{\theta' \sim Q_\theta} \widehat{\mathcal{L}}_S(f_{\theta'}) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2} - \frac{1}{2} y_i F_\theta(\mathbf{x}_i)\right].$$

Complexity term

$$\mathrm{KL}(Q_\theta \| P_{\theta_0}) = \frac{1}{2}\|\theta - \theta_0\|^2 .$$

Derivatives

Chain rule.

$$\frac{\partial \widehat{\mathcal{L}}_S(F_\theta)}{\partial \theta} = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial \ell(F_\theta(\mathbf{x}_i), y_i)}{\partial \theta} = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial F_\theta(\mathbf{x}_i)}{\partial \theta}\, \ell'(F_\theta(\mathbf{x}_i), y_i)\,,$$

Hidden layer partial derivatives.

$$\frac{\partial}{\partial \mathbf{w}_1^k} F_\theta(\mathbf{x}) = \frac{\mathbf{x}}{2^{\frac{3}{2}}\|\mathbf{x}\|} \mathrm{erf}'\left(\frac{\mathbf{w}_1^k \cdot \mathbf{x}}{\sqrt{2}\,\|\mathbf{x}\|}\right) \sum_{\mathbf{s}\in\{-1,1\}^{d_1}} s_k\, \mathrm{erf}\left(\frac{\mathbf{w}_2 \cdot \mathbf{s}}{\sqrt{2 d_1}}\right)\left[\frac{\Pr(\mathbf{s}|\mathbf{x}, \mathbf{W}_1)}{\Pr(s_k|\mathbf{x}, \mathbf{w}_1^k)}\right],$$

$$\text{with} \quad \mathrm{erf}'(x) := \tfrac{2}{\sqrt{\pi}} e^{-x^2}\,.$$

Stochastic Approximation

$$F_\theta(\mathbf{x}) = \sum_{\mathbf{s}\in\{-1,1\}^{d_1}} F_{\mathbf{w}_2}(\mathbf{s})\,\Pr(\mathbf{s}|\mathbf{x},\mathbf{W}_1)$$

Monte Carlo sampling

We generate T random binary vectors $\{\mathbf{s}^t\}_{t=1}^T$ according to $\Pr(\mathbf{s}|\mathbf{x},\mathbf{W}_1)$

Prediction.

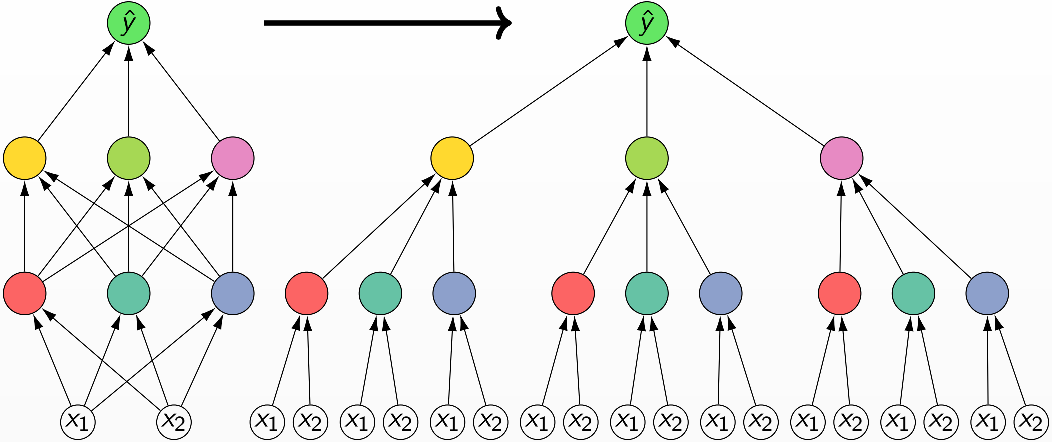
$$F_\theta(\mathbf{x}) \approx \frac{1}{T}\sum_{t=1}^{T} F_{\mathbf{w}_2}(\mathbf{s}^t)\,.$$

Derivatives.

$$\frac{\partial}{\partial \mathbf{w}_1^k} F_\theta(\mathbf{x}) \approx \frac{\mathbf{x}}{2^{\frac{3}{2}}\,\|\mathbf{x}\|}\,\mathrm{erf}'\left(\frac{\mathbf{w}_1^k\cdot\mathbf{x}}{\sqrt{2}\,\|\mathbf{x}\|}\right)\frac{1}{T}\sum_{t=1}^{T}\frac{s_k^t}{\Pr(s_k^t|\mathbf{x},\mathbf{w}_1^k)}F_{\mathbf{w}_2}(\mathbf{s}^t)\,.$$

Plan

More Layers



$$F_1^{(j)}(\mathbf{x}) = \mathrm{erf}\left(\frac{\mathbf{w}_1^j \cdot \mathbf{x}}{\sqrt{2}\|\mathbf{x}\|}\right), \qquad F_{k+1}^{(j)}(\mathbf{x}) = \sum_{\mathbf{s}\in\{-1,1\}^{d_k}} \mathrm{erf}\left(\frac{\mathbf{w}_{k+1}^j \cdot \mathbf{s}}{\sqrt{2d_k}}\right) \prod_{i=1}^{d_k} \left(\frac{1}{2} + \frac{1}{2} s_i \times F_k^{(i)}(\mathbf{x})\right)$$

Plan

Model name	Cost function	Train split	Valid split	Model selection	Prior
MLP–tanh	linear loss, L2 regularized	80%	20%	valid linear loss	-
PBGNet$_\ell$	linear loss, L2 regularized	80%	20%	valid linear loss	random init
PBGNet	**PAC-Bayes bound**	**100 %**	-	**PAC-Bayes bound**	**random init**
PBGNet$_{\text{pre}}$					
– pretrain	linear loss (20 epochs)	50%	-	-	random init
– final	PAC-Bayes bound	50%	-	PAC-Bayes bound	pretrain

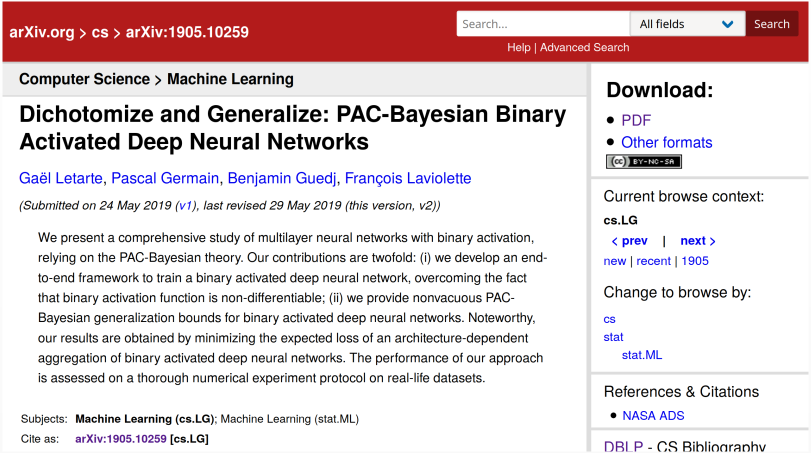
Dataset	MLP–tanh		PBGNet$_\ell$		PBGNet			PBGNet$_{\text{pre}}$		
	E_S	E_T	E_S	E_T	E_S	E_T	Bound	E_S	E_T	Bound
ads	0.021	0.037	0.018	**0.032**	0.024	0.038	0.283	0.034	0.033	0.058
adult	0.128	0.149	0.136	**0.148**	0.158	0.154	0.227	0.153	0.151	0.165
mnist17	0.003	**0.004**	0.008	0.005	0.007	0.009	0.067	0.003	0.005	0.009
mnist49	0.002	**0.013**	0.003	0.018	0.034	0.039	0.153	0.018	0.021	0.030
mnist56	0.002	0.009	0.002	0.009	0.022	0.026	0.103	0.008	**0.008**	0.017
mnistLH	0.004	**0.017**	0.005	0.019	0.071	0.073	0.186	0.026	0.026	0.033

Perspectives

- Transfer learning
- Multiclass
- CNN

Merci !

https://arxiv.org/abs/1905.10259



arXiv.org > cs > arXiv:1905.10259

Search... All fields Search

Help | Advanced Search

Computer Science > Machine Learning

Dichotomize and Generalize: PAC-Bayesian Binary Activated Deep Neural Networks

Gaël Letarte, Pascal Germain, Benjamin Guedj, François Laviolette

(Submitted on 24 May 2019 (v1), last revised 29 May 2019 (this version, v2))

We present a comprehensive study of multilayer neural networks with binary activation, relying on the PAC-Bayesian theory. Our contributions are twofold: (i) we develop an end-to-end framework to train a binary activated deep neural network, overcoming the fact that binary activation function is non-differentiable; (ii) we provide nonvacuous PAC-Bayesian generalization bounds for binary activated deep neural networks. Noteworthy, our results are obtained by minimizing the expected loss of an architecture-dependent aggregation of binary activated deep neural networks. The performance of our approach is assessed on a thorough numerical experiment protocol on real-life datasets.

Subjects: **Machine Learning (cs.LG)**; Machine Learning (stat.ML)

Cite as: **arXiv:1905.10259 [cs.LG]**

Download:

- PDF
- Other formats

(cc) BY-NC-SA

Current browse context:

cs.LG

< prev | next >

new | recent | 1905

Change to browse by:

cs

stat

stat.ML

References & Citations

- NASA ADS

DBLP - CS Bibliography

References

Shawe-Taylor, John et Robert C. Williamson (1997). “A PAC Analysis of a Bayesian Estimator”. In : *COLT*.

McAllester, David (1999). “Some PAC-Bayesian Theorems”. In : *Machine Learning* 37.3.

Catoni, Olivier (2003). “A PAC-Bayesian approach to adaptive classification”. In : *preprint* 840.

McAllester, David (2003). “PAC-Bayesian Stochastic Model selection”. In : *Machine Learning* 51.1.

Langford, John et Rich Caruana (2001). “(Not) Bounding the True Error”. In : *NIPS*. MIT Press, p. 809-816.

Dziugaite, Gintare Karolina et Daniel M. Roy (2017). “Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data”. In : *UAI*. AUAI Press.

Germain, Pascal et al. (2009). “PAC-Bayesian learning of linear classifiers”. In : *ICML*. ACM, p. 353-360.