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

Gaël Letarte[1], Pascal Germain[2], Benjamin Guedj[2,3], François Laviolette[1]

[1] Département d'informatique et de génie logiciel, Université Laval, Québec, Canada
[2] Équipe-projet Modal, Inria Lille - Nord Europe, Villeneuve d'Ascq, France
[3] UCL Centre for Artificial Intelligence, University College London, London, England

Introduction

We present a comprehensive study of multilayer neural networks with binary activation, relying on the PAC-Bayesian theory.

Contributions

- An end-to-end framework to train a binary activated deep neural network.
- Nonvacuous PAC-Bayesian generalization bounds for binary activated deep neural networks.

PAC-Bayesian Theory

Given a data distribution $\mathcal{D}$, a training set $S=\{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)\}\sim\mathcal{D}^n$, with $\mathbf{x}_i\in\mathbb{R}^{d_0}$ and $y_i\in\{-1,1\}$, a loss $\ell:\{-1,1\}^2\to[0,1]$, a predictor $f\in\mathcal{F}$:

$$\mathcal{L}_{\mathcal{D}}(f) = \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim\mathcal{D}} \ell\big(f(\mathbf{x}),y\big) \qquad \textit{generalization loss}$$

$$\widehat{\mathcal{L}}_S(f) = \frac{1}{n}\sum_{i=1}^{n}\ell\big(f(\mathbf{x}_i),y_i\big) \qquad \textit{empirical loss}$$

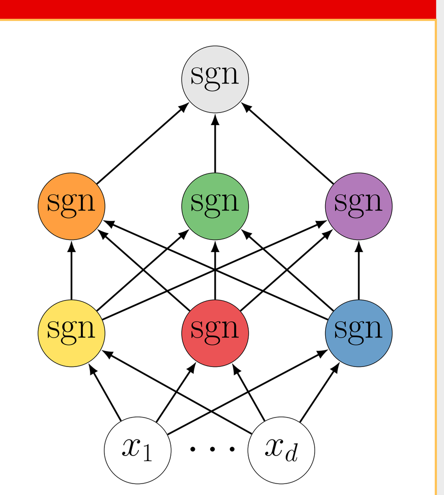
PAC-Bayesian Theorem

For $t>0$, for any prior P on $\mathcal{F}$, with probability $1-\delta$ on the choice of $S\sim\mathcal{D}^n$, we have for all posterior distribution P on $\mathcal{F}$:

$$\mathop{\mathbf{E}}_{f\sim\rho}\mathcal{L}_{\mathcal{D}}(f)\leq\mathop{\mathbf{E}}_{f\sim\rho}\widehat{\mathcal{L}}_S(f)+\sqrt{\frac{1}{2n}\left[\mathrm{KL}(Q\|P)+\ln\frac{2\sqrt{n}}{\delta}\right]}$$

Binary Activated Neural Networks

- L fully connected layers
- d_k denotes the number of neurons of the k^{th} layer
- $\mathrm{sgn}(a)=1$ if $a>0$ and $sgn(a)-1$ otherwise.
- Weights matrices : $\mathbf{W}_k\in\mathbb{R}^{d_k\times d_{k-1}}$, $\theta=\mathrm{vec}\big(\{\mathbf{W}_k\}_{k=1}^L\big)\in\mathbb{R}^D$.



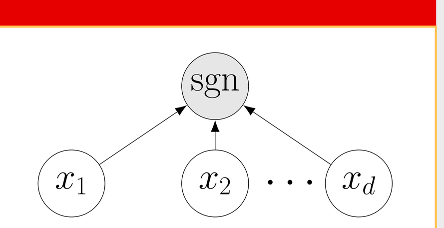
Prediction

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

Linear Classifier

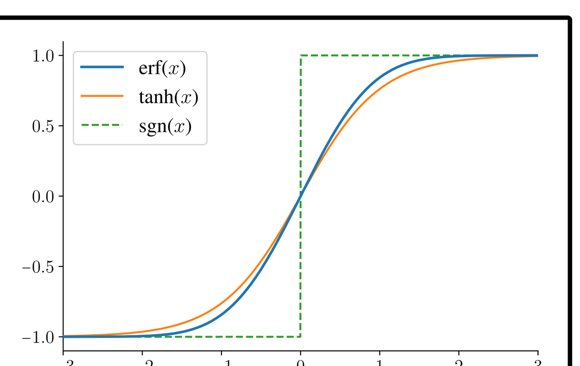
"PAC-Bayesian Learning of Linear Classifiers (Germain 2009)"

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



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



Bound minimization

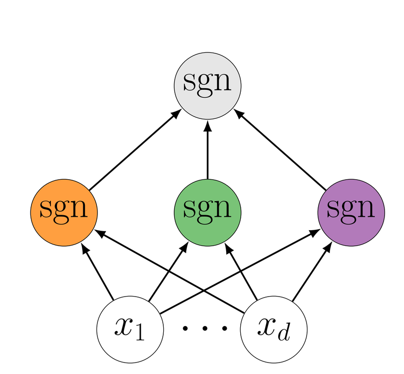
$$C\,n\,\widehat{\mathcal{L}}_S(F_{\mathbf{w}})+\mathrm{KL}(Q_{\mathbf{w}}\|P_{\mathbf{w}_0}) = C\tfrac{1}{2}\textstyle\sum_{i=1}^{n}\mathrm{erf}\left(-y_i\frac{\mathbf{w}\cdot\mathbf{x}_i}{\sqrt{d}\|\mathbf{x}_i\|}\right)+\tfrac{1}{2}\|\mathbf{w}-\mathbf{w}_0\|^2$$

Shallow Learning

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

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

$$\begin{aligned}F_\theta(\mathbf{x}) &= \mathop{\mathbf{E}}_{\tilde\theta\sim Q_\theta} f_{\tilde\theta}(\mathbf{x})\\ &= \int_{\mathbb{R}^{d_1\times d_0}}Q_1(\mathbf{V}_1)\int_{\mathbb{R}^{d_1}}Q_2(\mathbf{v}_2)\mathrm{sgn}(\mathbf{v}_2\cdot\mathrm{sgn}(\mathbf{V}_1\mathbf{x}))d\mathbf{v}_2d\mathbf{V}_1\\ &= \sum_{\mathbf{s}\in\{-1,1\}^{d_1}}\mathrm{erf}\left(\tfrac{\mathbf{w}_2\cdot\mathbf{s}}{\sqrt{2d_1}}\right)\int_{\mathbb{R}^{d_1\times d_0}}\mathbb{1}[\mathbf{s}=\mathrm{sgn}(\mathbf{V}_1\mathbf{x})]Q_1(\mathbf{V}_1)\,d\mathbf{V}_1\\ &= \sum_{\mathbf{s}\in\{-1,1\}^{d_1}}\underbrace{\mathrm{erf}\left(\frac{\mathbf{w}_2\cdot\mathbf{s}}{\sqrt{2d_1}}\right)}_{F_{\mathbf{w}_2}(\mathbf{s})}\underbrace{\prod_{i=1}^{d_1}\left[\frac{1}{2}+\frac{s_i}{2}\mathrm{erf}\left(\frac{\mathbf{w}_1^i\cdot\mathbf{x}}{\sqrt{2}\|\mathbf{x}\|}\right)\right]}_{\Pr(\mathbf{s}|\mathbf{x},\mathbf{W}_1)}\end{aligned}$$



PAC-Bayesian bound ingredients

- Empirical loss : $\widehat{\mathcal{L}}_S(F_\theta)=\mathbf{E}_{\theta'\sim Q_\theta}\widehat{\mathcal{L}}_S(f_{\theta'})=\frac{1}{n}\sum_{i=1}^n\left[\frac{1}{2}-\frac{1}{2}y_iF_\theta(\mathbf{x}_i)\right]$.
- Complexity term : $\mathrm{KL}(Q_\theta\|P_{\theta_0})=\frac{1}{2}\|\theta-\theta_0\|^2$
- Generalization bound : $\frac{1}{1-e^{-C}}\left(1-\exp\left(-C\,\widehat{\mathcal{L}}_S(F_\theta)-\frac{1}{n}[\mathrm{KL}(Q_\theta\|P_{\theta_0})+\ln\frac{2\sqrt{n}}{\delta}]\right)\right)$

Visualization

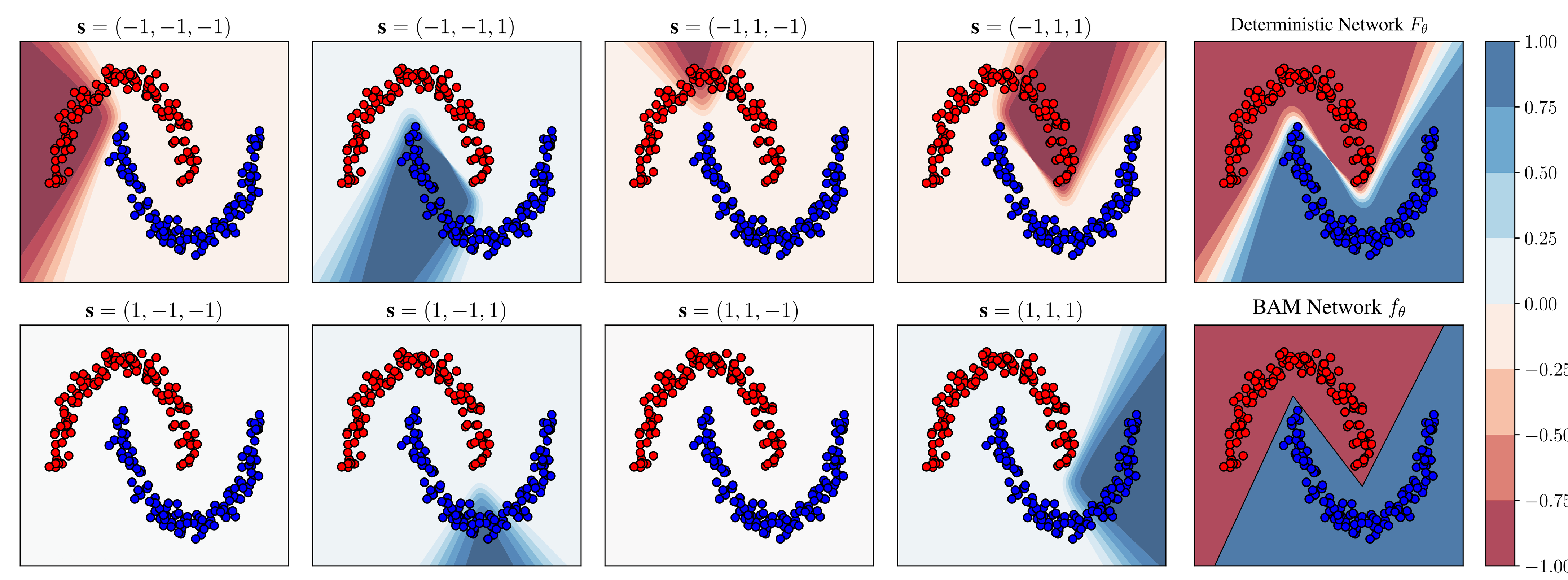


FIGURE 1: Illustration of the proposed method for a one hidden layer network of size d_1=3, interpreted as a majority vote over 8 binary representations $\mathbf{s}\in\{-1,1\}^3$. For each $\mathbf{s}$, a plot shows the values of $F_{\mathbf{w}_2}(\mathbf{s})\Pr(\mathbf{s}|\mathbf{x},\mathbf{W}_1)$.

Stochastic Approximation

Prediction.

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

Hidden layer partial derivatives, with $\mathrm{erf}'(x):=\frac{2}{\sqrt{\pi}}e^{-x^2}$.

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

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

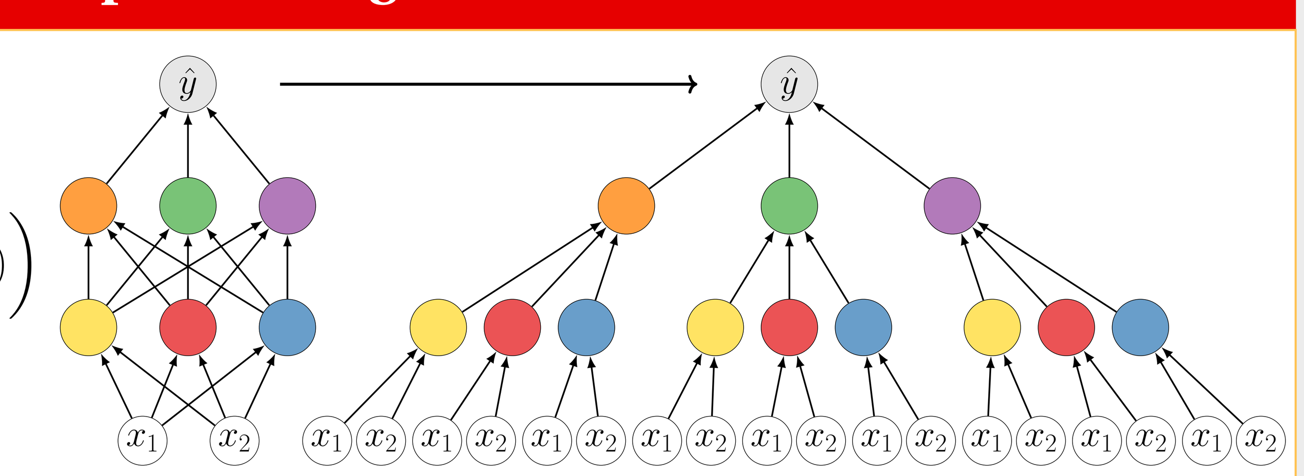
$$F_\theta(\mathbf{x})\approx\frac{1}{T}\sum_{t=1}^{T}F_{\mathbf{w}_2}(\mathbf{s}^t)$$

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

Deep Learning

$$F_1^{(j)}(\mathbf{x})=\mathrm{erf}\left(\frac{\mathbf{w}_1^j\cdot\mathbf{x}}{\sqrt{2}\|\mathbf{x}\|}\right)$$

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



Kullback-Leibler regularization

$$\mathrm{KL}\Big(Q_{\zeta(\theta)}\,\big\|\,P_{\zeta(\mu)}\Big)=\frac{1}{2}\left(\|\mathbf{w}_L-\mathbf{u}_L\|^2+\sum_{i=1}^{L-1}d_{k+1}^\dagger\|\mathbf{W}_i-\mathbf{U}_i\|_F^2\right)\text{ for }d_k^\dagger:=\textstyle\prod_{i=k}^{L}d_i\,.$$

Experiment

Results

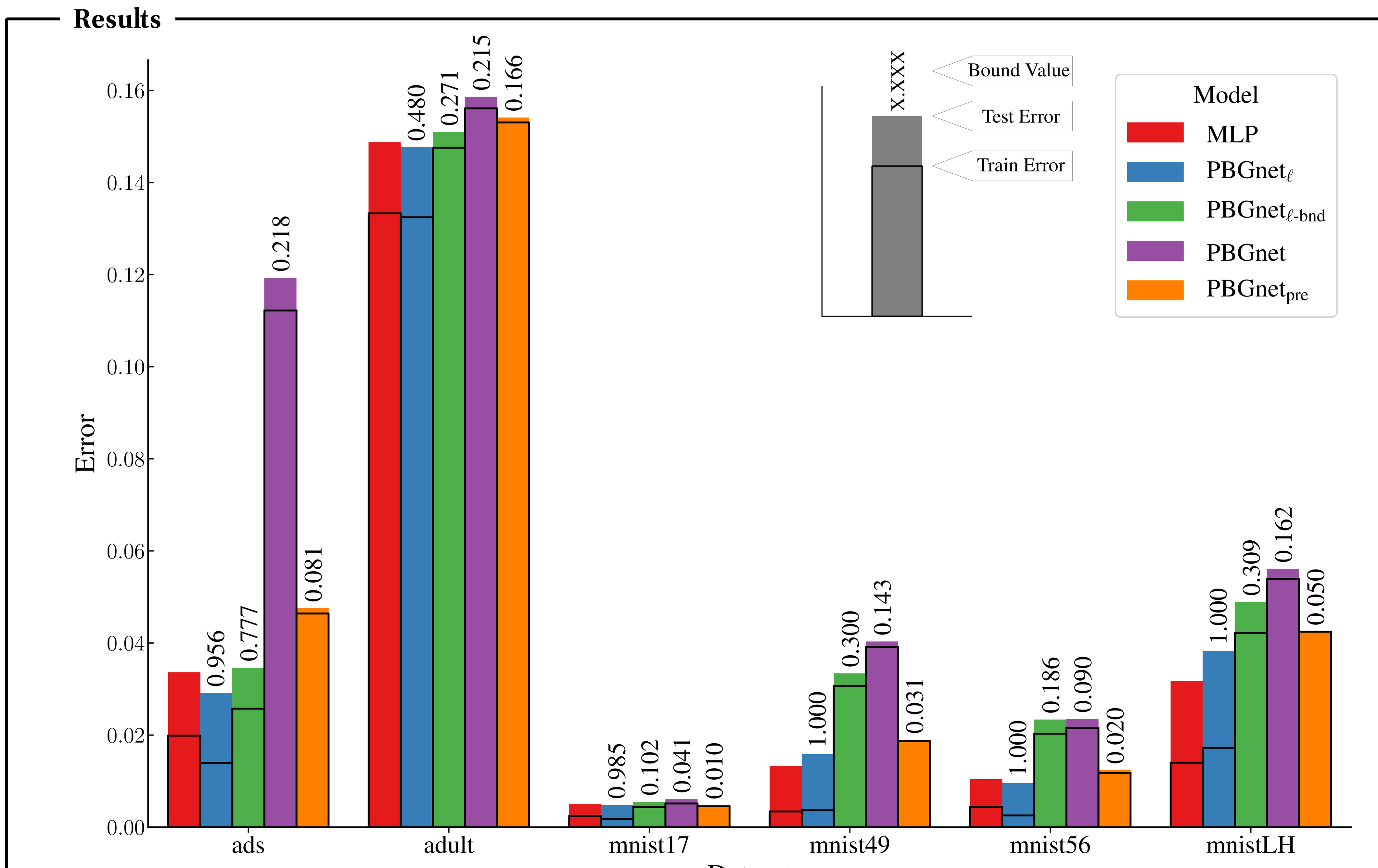


FIGURE 2: Experiment results for the considered models on the binary classification datasets. PAC-Bayesian bounds hold with probability 0.95.

Contact : gael.letarte.1@ulaval.ca **Code :** github.com/gletarte/dichotomize-and-generalize