An Equivalence between Bayesian Priors and Penalties in Variational Inference

ISBA 2022

Pierre Wolinski[1], Guillaume Charpiat[2], Yann Ollivier[3]

[1]Post-doc, UGA / Inria Grenoble, LJK, Statify Team (France)
[2]Inria Paris-Saclay, Tau Team (France)
[3]Facebook, Paris (France)

28/06/2022

Introduction

Why Bayesian neural networks?

Introduction

Why Bayesian neural networks?

- Neural Network (NN) $F_{\boldsymbol{\theta}}$, parameters $\boldsymbol{\theta}$;
- measure the incertitude on $\boldsymbol{\theta}$ after training
  $\Rightarrow$ Bayesian inference;

Introduction

Why Bayesian neural networks?

- Neural Network (NN) $F_{\boldsymbol{\theta}}$, parameters $\boldsymbol{\theta}$;
- measure the incertitude on $\boldsymbol{\theta}$ after training
  $\Rightarrow$ Bayesian inference;
- how to perform Bayesian inference on $\boldsymbol{\theta}$?
  $\Rightarrow$ Bayesian posterior intractable in general;

Introduction

Why Bayesian neural networks?

- Neural Network (NN) $F_{\boldsymbol{\theta}}$, parameters $\boldsymbol{\theta}$;
- measure the incertitude on $\boldsymbol{\theta}$ after training
  $\Rightarrow$ Bayesian inference;
- how to perform Bayesian inference on $\boldsymbol{\theta}$?
  $\Rightarrow$ Bayesian posterior intractable in general;
- possible solution: approximate the Bayesian posterior in a scalable way
  $\Rightarrow$ Variational Inference (VI);

Introduction

Why Bayesian neural networks?

- Neural Network (NN) $F_{\boldsymbol{\theta}}$, parameters $\boldsymbol{\theta}$;
- measure the incertitude on $\boldsymbol{\theta}$ after training
  $\Rightarrow$ Bayesian inference;
- how to perform Bayesian inference on $\boldsymbol{\theta}$?
  $\Rightarrow$ Bayesian posterior intractable in general;
- possible solution: approximate the Bayesian posterior in a scalable way
  $\Rightarrow$ Variational Inference (VI);
- variational posterior easily computable with standard deep learning libraries (TensorFlow, PyTorch).

Bayesian Framework

Setup.

- set $\mathcal{D}$ of data points (x_i, y_i) randomly sampled from $\mathbb{P}$;
  for convenience, let $\mathbf{x} = (x_i)_i$ and $\mathbf{y} = (y_i)_i$;

Bayesian Framework

Setup.

- set $\mathcal{D}$ of data points (x_i, y_i) randomly sampled from $\mathbb{P}$;
  for convenience, let $\mathbf{x} = (x_i)_i$ and $\mathbf{y} = (y_i)_i$;
- $F_\theta(x_i)$ outputs a distribution $p_\theta(\cdot|x_i)$ over y_i;

Bayesian Framework

Setup.

- set $\mathcal{D}$ of data points (x_i, y_i) randomly sampled from $\mathbb{P}$; for convenience, let $\mathbf{x} = (x_i)_i$ and $\mathbf{y} = (y_i)_i$;
- $F_{\boldsymbol{\theta}}(x_i)$ outputs a distribution $p_{\boldsymbol{\theta}}(\cdot|x_i)$ over y_i;
- prior distribution α on $\boldsymbol{\theta}$.

Bayesian Framework

Setup.

- set $\mathcal{D}$ of data points (x_i, y_i) randomly sampled from $\mathbb{P}$; for convenience, let $\mathbf{x} = (x_i)_i$ and $\mathbf{y} = (y_i)_i$;
- $F_{\boldsymbol{\theta}}(x_i)$ outputs a distribution $p_{\boldsymbol{\theta}}(\cdot|x_i)$ over y_i;
- prior distribution α on $\boldsymbol{\theta}$.

$\Rightarrow$ posterior $\pi_{\mathcal{D}}$:

$$\pi_{\mathcal{D}}(\boldsymbol{\theta}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})\,\alpha(\boldsymbol{\theta})}{\int p_{\boldsymbol{\theta}'}(\mathbf{y}|\mathbf{x})\,\alpha(\boldsymbol{\theta}')\,\mathrm{d}\boldsymbol{\theta}'}.$$

Bayesian Framework

Setup.

- set $\mathcal{D}$ of data points (x_i, y_i) randomly sampled from $\mathbb{P}$; for convenience, let $\mathbf{x} = (x_i)_i$ and $\mathbf{y} = (y_i)_i$;
- $F_{\boldsymbol{\theta}}(x_i)$ outputs a distribution $p_{\boldsymbol{\theta}}(\cdot|x_i)$ over y_i;
- prior distribution α on $\boldsymbol{\theta}$.

$\Rightarrow$ posterior $\pi_{\mathcal{D}}$:

$$\pi_{\mathcal{D}}(\boldsymbol{\theta}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})\,\alpha(\boldsymbol{\theta})}{\int p_{\boldsymbol{\theta}'}(\mathbf{y}|\mathbf{x})\,\alpha(\boldsymbol{\theta}')\,\mathrm{d}\boldsymbol{\theta}'}.$$

Usual approximation: MAP: $\boldsymbol{\theta}^*_{\mathrm{MAP}} = \arg\max \pi_{\mathcal{D}}(\boldsymbol{\theta})$.
(common in NNs)

Variational Framework

Variational inference.

- parameterized family $\mathcal{B} = \{\beta_\phi : \phi \in \Phi\}$;

Variational Framework

Variational inference.

- parameterized family $\mathcal{B} = \{\beta_\phi : \phi \in \Phi\}$;
- approximate the Bayesian posterior $\pi_\mathcal{D}$ by the closest distribution in $\mathcal{B}$:

$$\phi^* = \arg\min_{\phi \in \Phi} \mathrm{KL}(\beta_\phi \| \pi_\mathcal{D})$$

Variational Framework

Variational inference.

- parameterized family $\mathcal{B} = \{\beta_{\boldsymbol{\phi}} : \boldsymbol{\phi} \in \Phi\}$;
- approximate the Bayesian posterior $\pi_{\mathcal{D}}$ by the closest distribution in $\mathcal{B}$:

$$\begin{aligned} \boldsymbol{\phi}^* &= \underset{\boldsymbol{\phi}\in\Phi}{\arg\min}\, \mathrm{KL}(\beta_{\boldsymbol{\phi}} \| \pi_{\mathcal{D}}) \\ &= \underset{\boldsymbol{\phi}\in\Phi}{\arg\min} \left[\underbrace{-\mathbb{E}_{\boldsymbol{\theta}\sim\beta_{\boldsymbol{\phi}}} \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) + \mathrm{KL}(\beta_{\boldsymbol{\phi}} \| \alpha)}_{L_{\mathrm{VI}}(\boldsymbol{\phi})} \right] \end{aligned}$$

Variational Framework

Variational inference.

- parameterized family $\mathcal{B} = \{\beta_{\boldsymbol{\phi}} : \boldsymbol{\phi} \in \Phi\}$;
- approximate the Bayesian posterior $\pi_{\mathcal{D}}$ by the closest distribution in $\mathcal{B}$:

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}\in\Phi} \mathrm{KL}(\beta_{\boldsymbol{\phi}} \| \pi_{\mathcal{D}})$$
$$= \arg\min_{\boldsymbol{\phi}\in\Phi} \left[\underbrace{-\mathbb{E}_{\boldsymbol{\theta}\sim\beta_{\boldsymbol{\phi}}} \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) + \mathrm{KL}(\beta_{\boldsymbol{\phi}} \| \alpha)}_{L_{\mathrm{VI}}(\boldsymbol{\phi})} \right]$$

- loss L_{VI}: known also as the Evidence Lower BOund, ELBO;

Variational Framework

Variational inference.

- parameterized family $\mathcal{B} = \{\beta_{\boldsymbol{\phi}} : \boldsymbol{\phi} \in \Phi\}$;
- approximate the Bayesian posterior $\pi_{\mathcal{D}}$ by the closest distribution in $\mathcal{B}$:

$$\begin{aligned}\boldsymbol{\phi}^* &= \arg\min_{\boldsymbol{\phi}\in\Phi} \mathrm{KL}(\beta_{\boldsymbol{\phi}} \| \pi_{\mathcal{D}}) \\ &= \arg\min_{\boldsymbol{\phi}\in\Phi} \left[\underbrace{-\mathbb{E}_{\boldsymbol{\theta}\sim\beta_{\boldsymbol{\phi}}} \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) + \mathrm{KL}(\beta_{\boldsymbol{\phi}} \| \alpha)}_{L_{\mathrm{VI}}(\boldsymbol{\phi})} \right]\end{aligned}$$

- loss L_{VI}: known also as the Evidence Lower BOund, ELBO;
- variational posterior: $\beta_{\boldsymbol{\phi}^*}$.

Example

Ref.: ***Practical variational inference for neural networks***, Graves, 2011.

VI setup:

- parameters $\theta_i^k \sim \mathcal{N}(\mu_i^k, (\sigma_i^k)^2)$, independently drawn;

Example

Ref.: ***Practical variational inference for neural networks***, Graves, 2011.

VI setup:

- parameters $\theta_i^k \sim \mathcal{N}(\mu_i^k, (\sigma_i^k)^2)$, independently drawn;
- variational parameters to train: $\phi = (\mu_i^k, (\sigma_i^k)^2)_{k,i}$;

Example

Ref.: ***Practical variational inference for neural networks***, Graves, 2011.

VI setup:

- parameters $\theta_i^k \sim \mathcal{N}(\mu_i^k, (\sigma_i^k)^2)$, independently drawn;
- variational parameters to train: $\phi = (\mu_i^k, (\sigma_i^k)^2)_{k,i}$;
- prior for parameters $(\theta_i^k)_i$ of neuron k: $\alpha \sim \mathcal{N}(0, s_k^2)$, where $s_k^2 = 1/\#\{\text{inputs of neuron } k\}$.

Example

Ref.: ***Practical variational inference for neural networks***, Graves, 2011.

VI setup:

- parameters $\theta_i^k \sim \mathcal{N}(\mu_i^k, (\sigma_i^k)^2)$, independently drawn;
- variational parameters to train: $\phi = (\mu_i^k, (\sigma_i^k)^2)_{k,i}$;
- prior for parameters $(\theta_i^k)_i$ of neuron k: $\alpha \sim \mathcal{N}(0, s_k^2)$, where $s_k^2 = 1/\#\{\text{inputs of neuron } k\}$.

Training procedure for one step:

1. draw $\boldsymbol{\theta} \sim \beta_{\boldsymbol{\phi}}$;

Example

Ref.: ***Practical variational inference for neural networks***, Graves, 2011.

VI setup:

- parameters $\theta_i^k \sim \mathcal{N}(\mu_i^k, (\sigma_i^k)^2)$, independently drawn;
- variational parameters to train: $\boldsymbol{\phi} = (\mu_i^k, (\sigma_i^k)^2)_{k,i}$;
- prior for parameters $(\theta_i^k)_i$ of neuron k: $\alpha \sim \mathcal{N}(0, s_k^2)$, where $s_k^2 = 1/\#\{\text{inputs of neuron } k\}$.

Training procedure for one step:

1. draw $\boldsymbol{\theta} \sim \beta_{\boldsymbol{\phi}}$;
2. send a batch of training points to $F_{\boldsymbol{\theta}}$;

Example

Ref.: ***Practical variational inference for neural networks***, Graves, 2011.

VI setup:

- parameters $\theta_i^k \sim \mathcal{N}(\mu_i^k, (\sigma_i^k)^2)$, independently drawn;
- variational parameters to train: $\phi = (\mu_i^k, (\sigma_i^k)^2)_{k,i}$;
- prior for parameters $(\theta_i^k)_i$ of neuron k: $\alpha \sim \mathcal{N}(0, s_k^2)$, where $s_k^2 = 1/\#\{\text{inputs of neuron } k\}$.

Training procedure for one step:

1. draw $\boldsymbol{\theta} \sim \beta_{\boldsymbol{\phi}}$;
2. send a batch of training points to $F_{\boldsymbol{\theta}}$;
3. backpropagate the gradient of the loss L until ϕ;

Example

Ref.: ***Practical variational inference for neural networks***, Graves, 2011.

VI setup:

- parameters $\theta_i^k \sim \mathcal{N}(\mu_i^k, (\sigma_i^k)^2)$, independently drawn;
- variational parameters to train: $\boldsymbol{\phi} = (\mu_i^k, (\sigma_i^k)^2)_{k,i}$;
- prior for parameters $(\theta_i^k)_i$ of neuron k: $\alpha \sim \mathcal{N}(0, s_k^2)$, where $s_k^2 = 1/\#\{\text{inputs of neuron } k\}$.

Training procedure for one step:

1. draw $\boldsymbol{\theta} \sim \beta_{\boldsymbol{\phi}}$;
2. send a batch of training points to $F_{\boldsymbol{\theta}}$;
3. backpropagate the gradient of the loss L until $\boldsymbol{\phi}$;
4. update the variational parameters: $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \eta \nabla_{\boldsymbol{\phi}} L$.

Penalty–KL Equivalence: Framework

Ref.: ***Practical Bayesian Framework for Backpropagation Networks***, MacKay, 1992.

Difference between the usual framework and Variational Inference:

$$\text{usual:} \quad L_{\text{usual}}(\theta) = -\ln p_\theta(\mathbf{y}|\mathbf{x}) + r(\theta)$$

Penalty–KL Equivalence: Framework

Ref.: ***Practical Bayesian Framework for Backpropagation Networks***, MacKay, 1992.

Difference between the usual framework and Variational Inference:

$$\begin{aligned}
\text{usual:} \quad L_{\mathrm{usual}}(\boldsymbol{\theta}) &= -\ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) + r(\boldsymbol{\theta}) \\
\text{Variational Inference:} \quad L_{\mathrm{VI}}(\beta) &= \mathbb{E}_{\boldsymbol{\theta}\sim\beta}\Big[-\ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})\Big] + \mathrm{KL}(\beta\|\alpha)
\end{aligned}$$

Penalty–KL Equivalence: Framework

Ref.: ***Practical Bayesian Framework for Backpropagation Networks***, MacKay, 1992.

Difference between the usual framework and Variational Inference:

$$\begin{array}{rrcrcr} \text{usual:} & L_{\mathrm{usual}}(\boldsymbol{\theta}) & = & -\ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) & + & r(\boldsymbol{\theta}) \\ \text{Variational Inference:} & L_{\mathrm{VI}}(\beta) & = & \mathbb{E}_{\boldsymbol{\theta}\sim\beta}\Big[-\ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})\Big] & + & \mathrm{KL}(\beta\|\alpha) \end{array}$$

Variational Inference:

- optimize $\beta \in \mathcal{B} = \{\beta_{\boldsymbol{\phi}}, \boldsymbol{\phi} \in \mathbb{R}^P\}$;
- Bayesian prior α;
- β contains more information than $\boldsymbol{\theta}$: uncertainty...

Link between a Penalty and a Bayesian Prior

We assume that $\boldsymbol{\theta} \sim \beta$. We define, for a penalty r and a prior α:

$$\begin{aligned} L(\beta) &= -\mathbb{E}_{\boldsymbol{\theta}\sim\beta} \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) &+& \quad r(\beta) \\ L_{\mathrm{VI}}(\beta) &= -\mathbb{E}_{\boldsymbol{\theta}\sim\beta} \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) &+& \quad \mathrm{KL}(\beta\|\alpha). \end{aligned}$$

Link between a Penalty and a Bayesian Prior

We assume that $\boldsymbol{\theta} \sim \beta$. We define, for a penalty r and a prior α:

$$
\begin{aligned}
L(\beta) &= -\mathbb{E}_{\boldsymbol{\theta}\sim\beta} \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) + r(\beta) \\
L_{\mathrm{VI}}(\beta) &= -\mathbb{E}_{\boldsymbol{\theta}\sim\beta} \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) + \mathrm{KL}(\beta\|\alpha).
\end{aligned}
$$

Questions

Given r, does there exist a prior α such that for all β, $r(\beta) = \mathrm{KL}(\beta\|\alpha)$?

If so, is there a systematic way to compute α from r?

Link between a Penalty and a Bayesian Prior

We assume that $\boldsymbol{\theta} \sim \beta$. We define, for a penalty r and a prior α:

$$\begin{aligned} L(\beta) &= -\mathbb{E}_{\boldsymbol{\theta}\sim\beta} \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) &+& \quad r(\beta) \\ L_{\mathrm{VI}}(\beta) &= -\mathbb{E}_{\boldsymbol{\theta}\sim\beta} \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) &+& \quad \mathrm{KL}(\beta\|\alpha). \end{aligned}$$

Questions

Given r, does there exist a prior α such that for all β, $r(\beta) = \mathrm{KL}(\beta\|\alpha)$?

If so, is there a systematic way to compute α from r?

Main assumptions and notation:

- $\mathcal{B}$ is translation-invariant: $\beta_{\boldsymbol{\phi}}(\theta) = \beta_{\mu,\boldsymbol{\nu}}(\theta) = \beta_{\mathbf{0},\boldsymbol{\nu}}(\theta - \mu)$ (μ is the mean);
- notation: $r(\beta_{\boldsymbol{\phi}}) = r(\mu, \boldsymbol{\nu}) = r_{\boldsymbol{\nu}}(\mu)$.

Typically: μ represents the mean and $\boldsymbol{\nu}$ the variance of $\beta_{\mu,\boldsymbol{\nu}}$.

Main Theorem

Goal: given a function r, find a probability distribution α such that:

$$\exists K \in \mathbb{R} : \forall \phi \in \Phi, \qquad r(\phi) = \mathrm{KL}(\beta_\phi \| \alpha) + K. \tag{1}$$

Main Theorem

Goal: given a function r, find a probability distribution α such that:

$$\exists K \in \mathbb{R} : \forall \phi \in \Phi, \qquad r(\phi) = \mathrm{KL}(\beta_\phi \| \alpha) + K. \tag{1}$$

Definition 1

Let $A_\nu = -\mathrm{Ent}(\beta_{\mathbf{0},\nu})\mathbb{1} - \mathcal{F}^{-1}\left[\frac{\mathcal{F}r_\nu}{\mathcal{F}\check{\beta}_{\mathbf{0},\nu}}\right]$.

r fulfills condition $(\star) \Leftrightarrow \begin{cases} A_\nu \text{ does not depend on } \boldsymbol{\nu} \\ A \text{ is a function s.t. } \exp(A) \text{ integrates to } \kappa > 0 \end{cases}.$

Main Theorem

Goal: given a function r, find a probability distribution α such that:

$$\exists K \in \mathbb{R} : \forall \phi \in \Phi, \qquad r(\phi) = \mathrm{KL}(\beta_\phi \| \alpha) + K. \tag{1}$$

Definition 1

Let $A_\nu = -\mathrm{Ent}(\beta_{0,\nu})\mathbb{1} - \mathcal{F}^{-1}\left[\frac{\mathcal{F} r_\nu}{\mathcal{F}\check{\beta}_{0,\nu}}\right]$.

r fulfills condition $(\star)$ $\Leftrightarrow \begin{cases} A_\nu \text{ does not depend on } \boldsymbol{\nu} \\ A \text{ is a function s.t. } \exp(A) \text{ integrates to } \kappa > 0 \end{cases}$.

Theorem 2 (informal)

Equation (1) *has a solution* $\alpha \in \mathcal{T}$ $\Leftrightarrow$ r *fulfills* $(\star)$ *and* $\alpha = \frac{1}{\kappa}\exp(A)$.

Further details

We recall A_ν:

$$A_\nu = -\mathrm{Ent}(\beta_{0,\nu})\mathbb{1} - \mathcal{F}^{-1}\left[\frac{\mathcal{F}r_\nu}{\mathcal{F}\check{\beta}_{0,\nu}}\right]$$

- warning: typical choice for the penalty: $r_\nu(\mu) \propto \mu^2$
  $\Rightarrow r_\nu \notin \mathcal{L}^2$
- Fourier transform of r_ν: theory of distributions
  $\Rightarrow \mathcal{F}r_\nu \propto -\delta''$.

Further details

We recall A_ν:

$$A_\nu = -\mathrm{Ent}(\beta_{0,\nu})\mathbb{1} - \mathcal{F}^{-1}\left[\frac{\mathcal{F}r_\nu}{\mathcal{F}\check{\beta}_{0,\nu}}\right]$$

- warning: typical choice for the penalty: $r_\nu(\mu) \propto \mu^2$
  $\Rightarrow r_\nu \notin \mathcal{L}^2$
- Fourier transform of r_ν: theory of distributions
  $\Rightarrow \mathcal{F}r_\nu \propto -\delta''$.

Corollary (informal)

For a given weight w drawn from $\beta_{\mu,\sigma^2} = \mathcal{N}(\mu,\sigma^2)$. If $r_{a,b}(\beta_{\mu,\sigma^2}) = a(\sigma^2) + b(\sigma^2)\mu^2$ corresponds to a prior α, then there exists σ_0^2 such that:

$$\alpha = \mathcal{N}(0,\sigma_0^2), \qquad a(\sigma^2) = \frac{1}{2\sigma_0^2}, \qquad b(\sigma^2) = \frac{1}{2}\left[\frac{\sigma^2}{\sigma_0^2} + \ln\left(\frac{\sigma_0^2}{\sigma^2}\right) - 1\right].$$

Application: fixing the penalty factor λ

Ref.: ***Understanding the difficulty of training deep feedforward neural networks***, Glorot and Bengio, 2010.

How to fix the regularization factor λ? Loss: $L(\phi) = \ell(\phi) + \lambda\tilde{r}(\phi)$.

Application: fixing the penalty factor λ

Ref.: ***Understanding the difficulty of training deep feedforward neural networks***, Glorot and Bengio, 2010.

How to fix the regularization factor λ? Loss: $L(\phi) = \ell(\phi) + \lambda \tilde{r}(\phi)$.

Determine λ.
Penalty–prior equivalence: $r_\lambda = \lambda \tilde{r} \Rightarrow$ prior α_λ;
Glorot's initialization: $\mathrm{Var}(\alpha_\lambda) = 1/P_\ell \Rightarrow \lambda = \ldots$

Application: fixing the penalty factor λ

Ref.: ***Understanding the difficulty of training deep feedforward neural networks***, Glorot and Bengio, 2010.

How to fix the regularization factor λ? Loss: $L(\phi) = \ell(\phi) + \lambda\tilde{r}(\phi)$.

Determine λ.
Penalty–prior equivalence: $r_\lambda = \lambda\tilde{r} \Rightarrow$ prior α_λ;
Glorot's initialization: $\mathrm{Var}(\alpha_\lambda) = 1/P_\ell \Rightarrow \lambda = \ldots$

Key hypothesis: the prior and the initialization distribution must have the same variance.

Application: fixing the penalty factor λ

Ref.: ***Understanding the difficulty of training deep feedforward neural networks***, Glorot and Bengio, 2010.

How to fix the regularization factor λ? Loss: $L(\phi) = \ell(\phi) + \lambda\tilde{r}(\phi)$.

Determine λ.
Penalty–prior equivalence: $r_\lambda = \lambda\tilde{r} \Rightarrow$ prior α_λ;
Glorot's initialization: $\mathrm{Var}(\alpha_\lambda) = 1/P_\ell \Rightarrow \lambda = \dots$

Key hypothesis: the prior and the initialization distribution must have the same variance.

Example: Gaussian distributions with $\mathcal{L}^2$ penalty.
$\beta_\phi = \beta_{\mu,\sigma^2} \sim \mathcal{N}(\mu, \sigma^2)$ and $r_\lambda(\beta_{\mu,\sigma^2}) = \lambda\mu^2$.
With the corollary, we prove that $\alpha = \alpha_{\sigma_0^2} \sim \mathcal{N}(0, \sigma_0^2)$, so
α fulfills Glorot $\Leftrightarrow$ $\sigma_0^2 = \frac{1}{P_\ell}$.

Experimental Results

Tested architectures: simple convolutional NN (CVNN) and VGG19.
Complete penalty: $\lambda \sum_\ell \lambda_\ell r(\boldsymbol{\theta}_\ell)$, where $\boldsymbol{\theta}_\ell$ is the tensor of the ℓ-th layer.

Experiments:

- "usual setup": λ_ℓ is set to λ_{usual} (found by heuristics);
- "Bayesian setup": λ_ℓ is set to $\lambda_{\mathrm{Bayesian}}$ (see above);
- in both setups: grid search over $\lambda \Rightarrow (\lambda^*, \mathrm{acc}^*)$;
- in the Bayesian setup: λ should be theoretically equal to $\lambda_{\mathrm{Th}} = 1/\#[\text{training set}]$.

	$\|w\|_2^2$		$\|w\|_1$		$\|w\|_{2,1}$		$\|w^T\|_{2,1}$	
	CVNN	VGG	CVNN	VGG	CVNN	VGG	CVNN	VGG
$\mathrm{acc}^*_{\mathrm{usual}}$ (%)	$88.00 \pm .4$	$93.35 \pm .15$	$88.36 \pm .3$	$93.17 \pm .3$	$88.43 \pm .14$	$92.78 \pm .19$	$88.04 \pm .4$	$93.37 \pm .09$
$\mathrm{acc}^*_{\mathrm{Bayesian}}$	$88.69 \pm .12$	$93.48 \pm .09$	$88.41 \pm .3$	$92.89 \pm .2$	$88.67 \pm .09$	$92.35 \pm .18$	$88.32 \pm .16$	$93.03 \pm .15$
$\mathrm{acc}_{\mathrm{Bayesian}}$	$88.25 \pm .3$	$93.28 \pm .17$	$87.48 \pm .08$	$92.74 \pm .19$	$87.45 \pm .17$	$92.24 \pm .14$	$85.49 \pm .3$	$92.85 \pm .06$
$\lambda_{\mathrm{Th}}/\lambda^*$	$10^{0.5}$	10^1	10^1	10^1	10^2	10^2	$10^{1.5}$	10^1

Discussion

Main question: why is this apparently well-founded method to find λ suboptimal?

Discussion

Main question: why is this apparently well-founded method to find λ suboptimal?
$\Rightarrow$ see the general problem called "cold posterior effect"

Reference: *How good is the Bayes posterior in deep neural networks really?* Wenzel et al., 2020.

Discussion

Main question: why is this apparently well-founded method to find λ suboptimal?
$\Rightarrow$ see the general problem called "cold posterior effect"

Reference: *How good is the Bayes posterior in deep neural networks really?* Wenzel et al., 2020.

Relation prior-initialization: are we right to assume that these distributions should be equal?

Discussion

Main question: why is this apparently well-founded method to find λ suboptimal?
$\Rightarrow$ see the general problem called "cold posterior effect"

Reference: *How good is the Bayes posterior in deep neural networks really?* Wenzel et al., 2020.

Relation prior-initialization: are we right to assume that these distributions should be equal?

Message. Penalty-prior equivalence: depends on the penalty r and the structure of the variational family $\mathcal{B}$.

Discussion

Main question: why is this apparently well-founded method to find λ suboptimal?
$\Rightarrow$ see the general problem called "cold posterior effect"

Reference: *How good is the Bayes posterior in deep neural networks really?* Wenzel et al., 2020.

Relation prior-initialization: are we right to assume that these distributions should be equal?

Message. Penalty-prior equivalence: depends on the penalty r and the structure of the variational family $\mathcal{B}$.

Possible improvements:

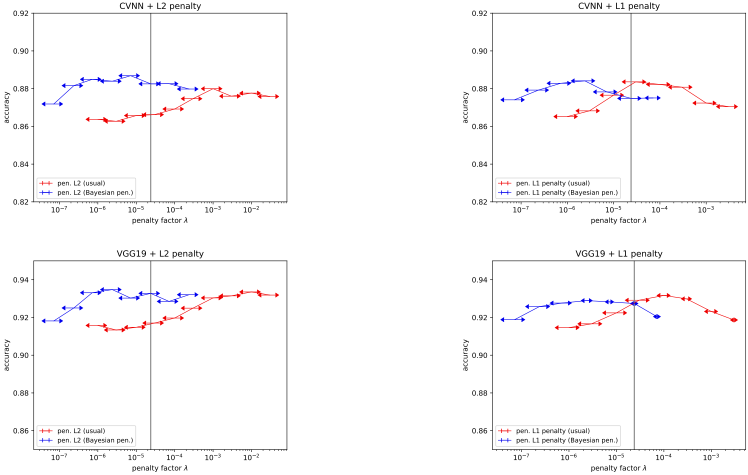
- improve the main theorem contribution;
- replace Glorot's initialization heuristics by the "Edge of Chaos"'s (more general).

Thank you!

References:

- *Practical Bayesian Framework for Backpropagation Networks*, MacKay, 1992;
- *An Introduction to Variational Methods for Graphical Models*, Jordan et al., 1999;
- *Understanding the difficulty of training deep feedforward neural networks*, Glorot and Bengio, 2010;
- *Practical variational inference for neural networks*, Graves, 2011;
- *Learning the number of neurons in deep networks*, Alvarez and Salzmann, 2016;
- *Deep information propagation*, Schoenholz et al., 2016;
- *Generalized variational inference: Three arguments for deriving new posteriors*, Knoblauch et al., 2019;
- *How good is the Bayes posterior in deep neural networks really?* Wenzel et al., 2020.

Penalty–KL Equivalence: Graphs (1)



Penalty–KL Equivalence: Graphs (2)