UNIVERSIDAD
DE GRANADA

Linearized Laplace Approximation for Modern Deep Learning (I)

Francisco Miguel Castro Macías
fcastro@ugr.es

Visual Information Processing Group (VIP)
Department of Computer Science and Artificial Intelligence (DECSAI)
University of Granada (UGR)

May 2, 2023

Overview

1. Introduction
   1.1 Overview
   1.2 Notation
   1.3 Laplace Approximation (LA)

2. Laplace Approximation for Bayesian Deep Learning
   2.1 Bayesian Deep Learning (BDL)
   2.2 BDL meets the LA
   2.3 BDL meets the Linearized Laplace Approximation (LLA)
   2.4 Model evidence for hyperparameter estimation

3. Adapting the Linearized Laplace Model Evidence for Modern Deep Learning
   3.1 Alternative adaptation of LLA
   3.2 Normalization layers

1. Introduction
   1.1 Overview
   1.2 Notation
   1.3 Laplace Approximation (LA)

2. Laplace Approximation for Bayesian Deep Learning
   2.1 Bayesian Deep Learning (BDL)
   2.2 BDL meets the LA
   2.3 BDL meets the Linearized Laplace Approximation (LLA)
   2.4 Model evidence for hyperparameter estimation

3. Adapting the Linearized Laplace Model Evidence for Modern Deep Learning
   3.1 Alternative adaptation of LLA
   3.2 Normalization layers

Our goal

$$\underbrace{\text{Sampling-based inference for large linear models, with application to}}_{\text{Coming soon}} \ \underbrace{\text{linearised Laplace}}_{\text{Today}}$$

We are going to follow

- **Original ideas.** Bayesian methods for adaptive models (Mackay, 1992).
- **Laplace Approximation and Linearized Laplace Approximation.** Improving predictions of Bayesian neural nets via local linearization, (Immer et al., 2021).
- **Linearized Laplace and Model Evidence.** Adapting the linearised laplace model evidence for modern deep learning, (Antorán et al., 2022a).
- **Linearized Laplace for large models.** Sampling-based inference for large linear models, with application to linearised Laplace, Antorán et al. (2022b).

Notation

Let $\mathbf{F} = (F_1, \ldots, F_M) \colon \mathbb{R}^N \to \mathbb{R}^M$ be differentiable and let $\mathrm{d}\mathbf{F}_{\mathbf{x}} \colon \mathbb{R}^N \to \mathbb{R}^M$ be its differential application at $\mathbf{x} \in \mathbb{R}^N$ given by $\mathrm{d}\mathbf{F}_{\mathbf{x}}(\mathbf{v}) = \partial_{\mathbf{x}}\mathbf{F}\mathbf{v}$ for each $\mathbf{v} \in \mathbb{R}^N$. Here, $\partial_{\mathbf{x}}\mathbf{F} \in \mathbb{R}^{M \times N}$ is the Jacobian matrix of $\mathbf{F}$ at $\mathbf{x}$ given by $(\partial_{\mathbf{x}}\mathbf{F})_{ij} = \frac{\partial F_j}{\partial x_i}(\mathbf{x})$. We denote $\nabla \mathbf{F}(\mathbf{x}) = (\partial_{\mathbf{x}}\mathbf{F})^\top \in \mathbb{R}^{N \times M}$.

When $M = 1$ and $\mathbf{F} = F$ is twice differentiable, the Hessian matrix of F at $\mathbf{x}$, denoted by $\nabla^2 F(\mathbf{x}) \in \mathbb{R}^{N \times N}$, is given by $(\nabla^2 F(\mathbf{x}))_{ij} = \frac{\partial^2 F}{\partial x_i \partial x_j}(\mathbf{x})$. If we consider the gradient as $\nabla F \colon \mathbb{R}^N \to \mathbb{R}^N$, then $\nabla^2 F(\mathbf{x}) = \partial_{\mathbf{x}}(\nabla F)^\top$.

Laplace Approximation (LA)

Let $h\colon \mathbb{R}^D \to \mathbb{R}$ be a sufficiently differentiable function. Consider its Taylor expansion around $\mathbf{a} \in \mathbb{R}^D$,

$$\begin{aligned} h(\mathbf{x}) &= h(\mathbf{a}) + (\mathbf{x}-\mathbf{a})^\top \nabla h(\mathbf{a}) + \frac{1}{2}(\mathbf{x}-\mathbf{a})^\top \nabla^2 h(\mathbf{a})(\mathbf{x}-\mathbf{a}) + O\left(\|\mathbf{x}-\mathbf{a}\|^2\right) \\ &\approx h(\mathbf{a}) + (\mathbf{x}-\mathbf{a})^\top \nabla h(\mathbf{a}) + \frac{1}{2}(\mathbf{x}-\mathbf{a})^\top \nabla^2 h(\mathbf{a})(\mathbf{x}-\mathbf{a}). \end{aligned}$$

Suppose that there exists $\mathbf{a}^* \in \mathbb{R}^D$ such that $\nabla h(\mathbf{a}^*) = \mathbf{0}$ (local extrema). Then,

$$h(\mathbf{x}) \approx h(\mathbf{a}^*) + \frac{1}{2}(\mathbf{x}-\mathbf{a}^*)^\top \nabla^2 h(\mathbf{a}^*)(\mathbf{x}-\mathbf{a}^*).$$

For densities. If $p\colon \mathbb{R}^D \to \mathbb{R}$ is a sufficiently differentiable density function with mode $\mathbf{a}^*$,

$$p(\mathbf{x}) \approx \mathcal{N}\left(\mathbf{x} \mid \mathbf{a}^*, \boldsymbol{\Sigma}\right), \quad \boldsymbol{\Sigma}^{-1} = -\nabla^2 \log p(\mathbf{a}^*).$$

1. Introduction
   1.1 Overview
   1.2 Notation
   1.3 Laplace Approximation (LA)

2. Laplace Approximation for Bayesian Deep Learning
   2.1 Bayesian Deep Learning (BDL)
   2.2 BDL meets the LA
   2.3 BDL meets the Linearized Laplace Approximation (LLA)
   2.4 Model evidence for hyperparameter estimation

3. Adapting the Linearized Laplace Model Evidence for Modern Deep Learning
   3.1 Alternative adaptation of LLA
   3.2 Normalization layers

Bayesian Deep Learning

Consider a dataset $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n) \colon n \in \{1, \dots, N\}\}$ where $\mathbf{x}_n \in \mathbb{R}^D$ and $\mathbf{y}_n \in \mathcal{Y}^C$ ($\mathcal{Y}$ can be $\mathbb{R}$ or $\{0, 1\}$). Let $\mathbf{f} = [f_1, \dots, f_C] : \mathbb{R}^D \times \mathbb{R}^P \to \mathbb{R}^C$ be a neural network that is trained to minimize the following loss function with respect to $\boldsymbol{\theta} \in \mathbb{R}^P$,

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{A}, \mathbf{f}) = \sum_n \ell\left(\mathbf{y}_n, \mathbf{f}(\mathbf{x}_n, \boldsymbol{\theta})\right) + R(\boldsymbol{\theta}, \mathbf{A}), \quad \widetilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathbf{A}, \mathbf{f}),$$

where ℓ is a negative log-likelihood and $\mathbf{A}$ denotes the regularization hyperparameters. Let us cast this under the Bayesian framework. Consider

$$p(\boldsymbol{\theta}, \mathcal{D}, \mathbf{A}; \mathbf{f}) = p(\mathbf{A}) p(\boldsymbol{\theta} \mid \mathbf{A}) \prod_n p(\mathbf{y}_n \mid \mathbf{f}(\mathbf{x}_n, \boldsymbol{\theta})) \prod_n p(\mathbf{x}_n),$$

$$p(\boldsymbol{\theta} \mid \mathbf{A}) = \exp(-R(\boldsymbol{\theta}, \mathbf{A}))/Z_R(\mathbf{A}), \quad p(\mathbf{y}_n \mid \mathbf{f}(\mathbf{x}_n, \boldsymbol{\theta})) = \exp(-\ell\left(\mathbf{y}_n, \mathbf{f}(\mathbf{x}_n, \boldsymbol{\theta})\right))/Z_\ell.$$

Then $p(\boldsymbol{\theta} \mid \mathcal{D}, \mathbf{A}; \mathbf{f}) \propto \exp\left(-\mathcal{L}(\boldsymbol{\theta}, \mathbf{A}, \mathbf{f})\right)$. Therefore, minimizing $\mathcal{L}(\boldsymbol{\theta}, \mathbf{A}, \mathbf{f})$ is equivalent to maximizing $\log p(\boldsymbol{\theta} \mid \mathcal{D}, \mathbf{A}; \mathbf{f})$. In this section, we will omit the dependence on $\mathbf{A}$.

BDL meets the LA

We make (probabilistic) predictions for a new input $\mathbf{x}^*$ using the posterior predictive

$$p(\mathbf{y}^* \mid \mathbf{x}^*, \mathcal{D}; \mathbf{f}) = \int_{\mathbb{R}^P} p(\mathbf{y}^* \mid \mathbf{f}(\mathbf{x}^*, \boldsymbol{\theta})) p(\boldsymbol{\theta} \mid \mathcal{D}; \mathbf{f}) \mathrm{d}\boldsymbol{\theta} = \mathbb{E}_{p(\boldsymbol{\theta} \mid \mathcal{D}; \mathbf{f})} \left[p(\mathbf{y}^* \mid \mathbf{f}(\mathbf{x}^*, \boldsymbol{\theta})) \right].$$

Computing $p(\boldsymbol{\theta} \mid \mathcal{D}; \mathbf{f})$ is, in practice, infeasible.

Laplace approximation (LA).

Assume that $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\widetilde{\boldsymbol{\theta}}, \mathbf{f}) = \mathbf{0}$. When applying the Laplace approximation, we obtain

$$p(\boldsymbol{\theta} \mid \mathcal{D}; \mathbf{f}) \approx \mathcal{N}\left(\boldsymbol{\theta} \mid \widetilde{\boldsymbol{\theta}}, \boldsymbol{\Sigma}\right),$$

$$\boldsymbol{\Sigma}^{-1} = \nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\widetilde{\boldsymbol{\theta}}, \mathbf{f}) = -\nabla^2_{\boldsymbol{\theta}} \log p(\widetilde{\boldsymbol{\theta}} \mid \mathcal{D}; \mathbf{f}) = \nabla^2_{\boldsymbol{\theta}} R(\widetilde{\boldsymbol{\theta}}) + \sum_n \nabla^2_{\boldsymbol{\theta}} \ell\left(\mathbf{y}_n, \mathbf{f}(\mathbf{x}_n, \widetilde{\boldsymbol{\theta}})\right).$$

BDL meets the LA: Problems

Observe that, for $\boldsymbol{\theta} \in \mathbb{R}^P$,

$$\nabla^2_{\boldsymbol{\theta}} \ell\left(\mathbf{y}_n, \mathbf{f}(\mathbf{x}_n, \boldsymbol{\theta})\right) = \partial_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})^\top \boldsymbol{\Lambda}_{\boldsymbol{\theta}}(\mathbf{x}_n, \mathbf{y}_n) \partial_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) + \sum_c \nabla^2_{\boldsymbol{\theta}} f_c(\mathbf{x}_n, \boldsymbol{\theta}) \frac{\partial}{\partial f_c} \ell(\mathbf{y}_n, \mathbf{f}(x_n, \theta)),$$

$$\boldsymbol{\Lambda}_{\boldsymbol{\theta}}(\mathbf{x}_n, \mathbf{y}_n) = \nabla^2_{\mathbf{f}} \ell(\mathbf{y}_n, \mathbf{f}) \Big|_{\mathbf{f}=\mathbf{f}(\mathbf{x}_n, \boldsymbol{\theta})}$$

This raises two problems:

- Calculation of the network Hessian $\nabla^2_{\boldsymbol{\theta}} f_c(\mathbf{x}_n, \boldsymbol{\theta})$ is computationally very expensive.
- Thus defined, $\nabla^2_{\boldsymbol{\theta}} \ell\left(\mathbf{y}_n, \mathbf{f}(\mathbf{x}_n, \boldsymbol{\theta})\right)$ is not positive semi-definite. Therefore, neither will $\boldsymbol{\Sigma}$.

BDL meets the LA: Generalized Gauss-Newton

The **Generalized Gauss Newton (GGN)** approximation becomes

$$\nabla^2_{\boldsymbol{\theta}} \ell\left(\mathbf{y}_n, \mathbf{f}(\mathbf{x}_n, \boldsymbol{\theta})\right) \approx \partial_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})^\top \boldsymbol{\Lambda}_{\boldsymbol{\theta}}(\mathbf{x}_n, \mathbf{y}_n) \partial_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}), \quad \forall \boldsymbol{\theta} \in \mathbb{R}^P$$

Where does this comes from? Consider the *local linearization of* $\mathbf{f}$ *around* $\widetilde{\boldsymbol{\theta}}$,

$$\mathbf{f}^{\text{lin}}(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{f}(\mathbf{x}, \widetilde{\boldsymbol{\theta}}) + \partial_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}, \widetilde{\boldsymbol{\theta}})(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}).$$

It holds that

$$\nabla^2_{\boldsymbol{\theta}} \ell\left(\mathbf{y}_n, \mathbf{f}^{\text{lin}}(\mathbf{x}_n, \boldsymbol{\theta})\right) \Big|_{\boldsymbol{\theta}=\widetilde{\boldsymbol{\theta}}} = \partial_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}_n, \widetilde{\boldsymbol{\theta}})^\top \boldsymbol{\Lambda}_{\widetilde{\boldsymbol{\theta}}}(\mathbf{x}_n, \mathbf{y}_n) \partial_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}_n, \widetilde{\boldsymbol{\theta}}).$$

Also, if $\widetilde{\boldsymbol{\theta}}$ is a local minimum of $\mathcal{L}(\widetilde{\boldsymbol{\theta}}, \mathbf{f})$, then it is also a local minimum of $\mathcal{L}(\widetilde{\boldsymbol{\theta}}, \mathbf{f}^{\text{lin}})$.

BDL meets the LA: Linearized Laplace

Linearized Laplace Approximation (LLA), Laplace Generalized Gauss Newton (Lap-GGN)

$$p(\boldsymbol{\theta} \mid \mathcal{D}; \mathbf{f}) \approx p(\boldsymbol{\theta} \mid \mathcal{D}; \mathbf{f}^{\text{lin}}) \approx \mathcal{N}\left(\boldsymbol{\theta} \mid \widetilde{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\text{GGN}}\right),$$

$$\boldsymbol{\Sigma}_{\text{GGN}}^{-1} = \nabla^2_{\boldsymbol{\theta}} R(\widetilde{\boldsymbol{\theta}}) + \sum_n \partial_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}_n, \widetilde{\boldsymbol{\theta}})^\top \boldsymbol{\Lambda}_{\widetilde{\boldsymbol{\theta}}}(\mathbf{x}_n, \mathbf{y}_n) \partial_{\boldsymbol{\theta}} \mathbf{f}(\mathbf{x}_n, \widetilde{\boldsymbol{\theta}}).$$

Important remarks:

- **Laplace Approximation (LA)**. Only one approximation: LA.
- **Linearized Laplace Approximation (LLA)**. Two consecutive approximations: (i) $\mathbf{f}(\cdot, \theta) \approx \mathbf{f}^{\text{lin}}(\cdot, \theta)$, (ii) LA. It is equivalent to (i) LA, (ii) GGN.
- **Keep in mind.** The approximation $\mathbf{f}(\cdot, \boldsymbol{\theta}) \approx \mathbf{f}^{\text{lin}}(\cdot, \boldsymbol{\theta})$ turns the underlying probabilistic model from a Bayesian Neural Network to a Gaussian Linear Model (GLM).

BDL meets the LLA: Making predictions

The LLA posterior corresponds to the posterior of the linearized model. Therefore, we should use this model to make predictions.

$$p(\mathbf{y}^* \mid \mathbf{x}^*, \mathcal{D}) \approx \mathbb{E}_{\mathcal{N}\left(\boldsymbol{\theta} \mid \widetilde{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\text{GGN}}\right)} \left[p(\mathbf{y}^* \mid \mathbf{f}^{\text{lin}}(\mathbf{x}^*, \boldsymbol{\theta})) \right].$$

Note that, if $\boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{\theta} \mid \widetilde{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\text{GGN}}\right)$, then $\mathbf{f}^{\text{lin}}(\cdot, \boldsymbol{\theta}) \sim \mathcal{N}\left(\mathbf{f}(\cdot, \widetilde{\boldsymbol{\theta}}), \partial_{\boldsymbol{\theta}}\mathbf{f}(\cdot, \widetilde{\boldsymbol{\theta}})^\top \boldsymbol{\Sigma}_{\text{GGN}} \partial_{\boldsymbol{\theta}}\mathbf{f}(\cdot, \widetilde{\boldsymbol{\theta}})\right)$. Therefore,

$$p(\mathbf{y}^* \mid \mathbf{x}^*, \mathcal{D}) \approx \mathbb{E}_{\hat{\mathbf{f}} \sim \mathbf{f}^{\text{lin}}(\mathbf{x}^*, \boldsymbol{\theta})} \left[p\left(\mathbf{y}^* \mid \hat{\mathbf{f}}\right) \right].$$

Remark. Previous works used BNN model along with the GGN approximation, which was shown to severely underfit.

Model evidence for hyperparameter estimation

Let us assume that the regularizer term corresponds to the weight decay approach, $R(\boldsymbol{\theta}, \mathbf{A}) = \frac{1}{2}\boldsymbol{\theta}^\top \mathbf{A}\boldsymbol{\theta}$, which leads to $p(\boldsymbol{\theta} \mid \mathbf{A}) = \mathcal{N}\left(\boldsymbol{\theta} \mid \mathbf{0}, \mathbf{A}^{-1}\right)$ and $\nabla^2_{\boldsymbol{\theta}} R(\boldsymbol{\theta}, \mathbf{A}) = \mathbf{A}$.

How to estimate it? We choose $\mathbf{A}$ as the most likely to generate the observed data given $\mathbf{f}^{\text{lin}}(\cdot, \boldsymbol{\theta})$, with $\boldsymbol{\theta} \sim \mathrm{p}(\boldsymbol{\theta} \mid \mathbf{A})$. Assuming $p(\mathbf{A})$ is uniform, this corresponds to maximizing the model evidence

$$\log p(\mathcal{D} \mid \mathbf{A}; \mathbf{f}^{\text{lin}}) = \log \int_{\mathbb{R}^p} p(\mathcal{D} \mid \boldsymbol{\theta}; \mathbf{f}^{\text{lin}}) p(\boldsymbol{\theta} \mid \mathbf{A}) \mathrm{d}\boldsymbol{\theta} = \log \int_{\mathbb{R}^p} \frac{\exp\left(-\mathcal{L}(\boldsymbol{\theta}, \mathbf{A}, \mathbf{f}^{\text{lin}})\right)}{Z_R(\mathbf{A})} \mathrm{d}\boldsymbol{\theta} + \text{const.}$$

The LA $\mathcal{L}(\boldsymbol{\theta}, \mathbf{A}, \mathbf{f}^{\text{lin}}) \approx \mathcal{L}(\widetilde{\boldsymbol{\theta}}, \mathbf{A}, \mathbf{f}^{\text{lin}}) + \frac{1}{2}\left(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}\right)^\top \boldsymbol{\Sigma}^{-1}_{\text{GGN}} \left(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}\right)$ leads to the objective

$$\log p(\mathcal{D} \mid \mathbf{A}; \mathbf{f}^{\text{lin}}) \approx \underbrace{-\frac{1}{2}\left[\widetilde{\boldsymbol{\theta}}^\top \mathbf{A}\widetilde{\boldsymbol{\theta}} + \log\det\left(\mathbf{A}^{-1}\boldsymbol{\Sigma}^{-1}_{\text{GGN}}\right)\right]}_{\mathcal{M}(\mathbf{A}, \widetilde{\boldsymbol{\theta}})} + \text{const}$$

1. Introduction
   1.1 Overview
   1.2 Notation
   1.3 Laplace Approximation (LA)

2. Laplace Approximation for Bayesian Deep Learning
   2.1 Bayesian Deep Learning (BDL)
   2.2 BDL meets the LA
   2.3 BDL meets the Linearized Laplace Approximation (LLA)
   2.4 Model evidence for hyperparameter estimation

3. Adapting the Linearized Laplace Model Evidence for Modern Deep Learning
   3.1 Alternative adaptation of LLA
   3.2 Normalization layers

Our assumptions do not match reality

Previous derivations assume that $\widetilde{\boldsymbol{\theta}}$ is a local minimum of $\mathcal{L}(\boldsymbol{\theta}, \mathbf{A}, \mathbf{f})$. We heavily rely on this to

- Approximate the posterior $p(\boldsymbol{\theta} \mid \mathcal{D}; \mathbf{f})$.
- Conclude that $\widetilde{\boldsymbol{\theta}}$ is also a local minimum of $\mathcal{L}(\boldsymbol{\theta}, \mathbf{A}, \mathbf{f}^{\text{in}})$ and approximate the posterior $p(\boldsymbol{\theta} \mid \mathcal{D}; \mathbf{f}^{\text{in}})$.
- Obtain the model evidence approximation $\mathcal{M}(\mathbf{A}, \widetilde{\boldsymbol{\theta}})$.

In practice, we monitor some separate validation metric to stop de training procedure, which may not lead to local minima. Also, normalization layers complicate the problem of minimizing $\mathcal{L}(\boldsymbol{\theta}, \mathbf{A}, \mathbf{f})$.

Alternative adaptation of LLA

Objective. Obtain a pair $(\boldsymbol{\theta}^*, \mathbf{A}^*)$ that satisfies

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathbf{A}^*, \mathbf{f}^{\text{in}}), \quad \mathbf{A}^* \in \arg\max_{\mathbf{A}} \mathcal{M}(\mathbf{A}, \boldsymbol{\theta}^*).$$

1. Start with an initial guess of $\mathbf{A}^*$ and minimize $\mathcal{L}(\boldsymbol{\theta}, \mathbf{A}^*, \mathbf{f})$ to obtain $\widetilde{\boldsymbol{\theta}}$.
2. Minimize $\mathcal{L}(\boldsymbol{\theta}, \mathbf{A}^*, \mathbf{f}^{\text{in}})$ obtaining $\boldsymbol{\theta}^*$. In most situations, $\boldsymbol{\theta} \mapsto \mathcal{L}(\boldsymbol{\theta}, \mathbf{A}, \mathbf{f}^{\text{in}})$ will be convex, yielding a minimizer (convergence guaranteed).
3. Maximize $\mathcal{M}(\mathbf{A}, \boldsymbol{\theta}^*)$ obtaining $\mathbf{A}^*$. For every $\boldsymbol{\theta}$, $\mathbf{A} \mapsto \mathcal{M}(\mathbf{A}, \boldsymbol{\theta})$ is concave, yielding a maximizer (convergence guaranteed).
4. If $(\boldsymbol{\theta}^*, \mathbf{A}^*)$ is not a stationary point, move to step 2.

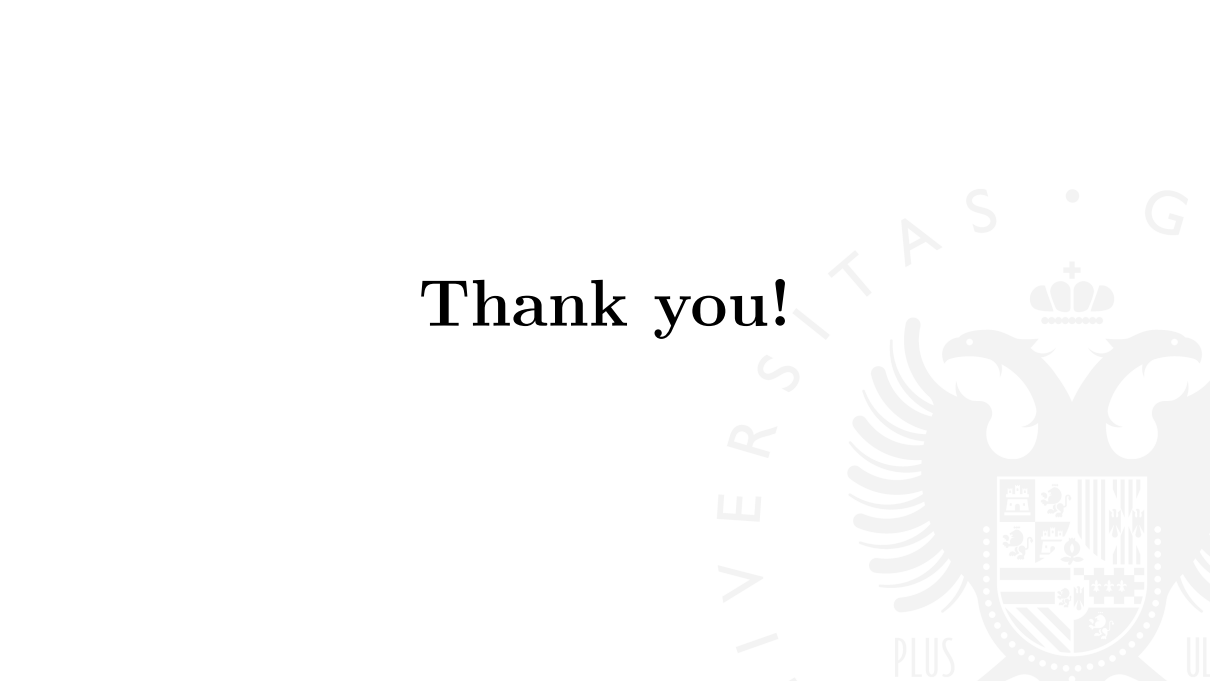
Finally, use the pair $(\boldsymbol{\theta}^*, \mathbf{A}^*)$ to compute the posterior approximation.

Normalization layers

Proposition (Antorán et al. (2022a))**.** When using weight decay, for any normalized network $\mathbf{f}$ and positive definite matrix $\mathbf{A}$, the loss $\mathcal{L}(\boldsymbol{\theta}, \mathbf{A}, \mathbf{f})$ has no local minima.

Recommendation (Antorán et al. (2022a))**.** When using the linearized Laplace method with a normalized network, use an independent regularizer for each normalized parameter group present.

Thank you!



References I

Javier Antorán, David Janz, James U Allingham, Erik Daxberger, Riccardo Rb Barbano, Eric Nalisnick, and José Miguel Hernández-Lobato. Adapting the linearised laplace model evidence for modern deep learning. In *International Conference on Machine Learning*, pages 796–821. PMLR, 2022a.

Javier Antorán, Shreyas Padhy, Riccardo Barbano, Eric Nalisnick, David Janz, and José Miguel Hernández-Lobato. Sampling-based inference for large linear models, with application to linearised laplace. *arXiv preprint arXiv:2210.04994*, 2022b.

Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of bayesian neural nets via local linearization. In *International Conference on Artificial Intelligence and Statistics*, pages 703–711. PMLR, 2021.

David John Cameron Mackay. *Bayesian methods for adaptive models*. California Institute of Technology, 1992.