International Conference on Machine Learning, 2022

Adapting the Linearised Laplace Model Evidence for Modern Deep Learning

Javier Antorán, James Allingham, David Janz, Erik Daxberger, Riccardo Barbano,
Eric Nalisnick, José Miguel Hernández-Lobato

UNIVERSITY OF CAMBRIDGE

Thank you to my collaborators!

James Allingham

David Janz

Erik Daxberger

Riccardo Barbano

Eric Nalisnick

José Miguel
Hernández-Lobato

Summary

- We identify pathologies in the linearised Laplace model evidence when applied to modern NNs
- We provide an adapted methodology that fixes these issues

Brief overview of linearised Laplace

Brief overview of linearised Laplace

1. Train a NN f to find a weight setting: $\tilde{\theta} \in \text{argmin}_{\theta}\, L(\theta) + ||\theta||_{\Lambda}^{2}$

Brief overview of linearised Laplace

1. Train a NN f to find a weight setting: $\tilde{\theta} \in \text{argmin}_\theta \, L(\theta) + ||\theta||_\Lambda^2$

Brief overview of linearised Laplace

1. Train a NN f to find a weight setting: $\tilde{\theta} \in \text{argmin}_{\theta} L(\theta) + ||\theta||^2_{\Lambda}$

2. Taylor expand f and loss around $\tilde{\theta}$ to obtain a **conjugate Gaussian-linear model**:

Brief overview of linearised Laplace

1. Train a NN f to find a weight setting: $\tilde{\theta} \in \text{argmin}_{\theta} L(\theta) + ||\theta||^2_{\Lambda}$

2. Taylor expand f and loss around $\tilde{\theta}$ to obtain a **conjugate Gaussian-linear model**:

   - Closed form predictive uncertainty

Brief overview of linearised Laplace

1. Train a NN f to find a weight setting: $\tilde{\theta} \in \text{argmin}_\theta \, L(\theta) + ||\theta||^2_\Lambda$

2. Taylor expand f and loss around $\tilde{\theta}$ to obtain a **conjugate Gaussian-linear model**:
   - Closed form predictive uncertainty
   - Closed form model evidence

Brief overview of linearised Laplace

1. Train a NN f to find a weight setting: $\tilde{\theta} \in \text{argmin}_\theta\, L(\theta) + ||\theta||^2_\Lambda$

2. Taylor expand f and loss around $\tilde{\theta}$ to obtain a **conjugate Gaussian-linear model**:

   - Closed form predictive uncertainty
   - Closed form model evidence $\longrightarrow$ Choose prior precision hyperparameter Λ

Issue 1: linearisation point $\tilde{\theta}$ is not a mode of the loss

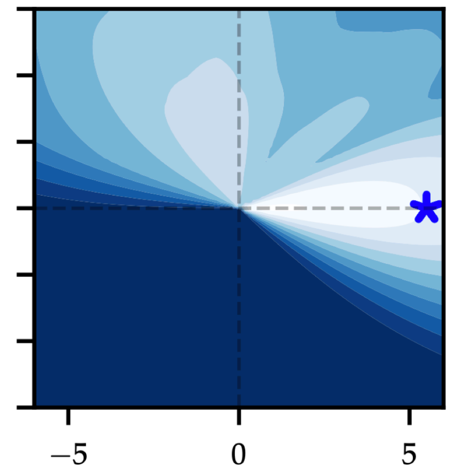
Issue 1: linearisation point $\tilde{\theta}$ is not a mode of the loss

- Stochastic optimisation, early stopping or normalisation layers prevent us from identifying a mode of the loss

Issue 1: linearisation point $\tilde{\theta}$ is not a mode of the loss

- Stochastic optimisation, early stopping or normalisation layers prevent us from identifying a mode of the loss

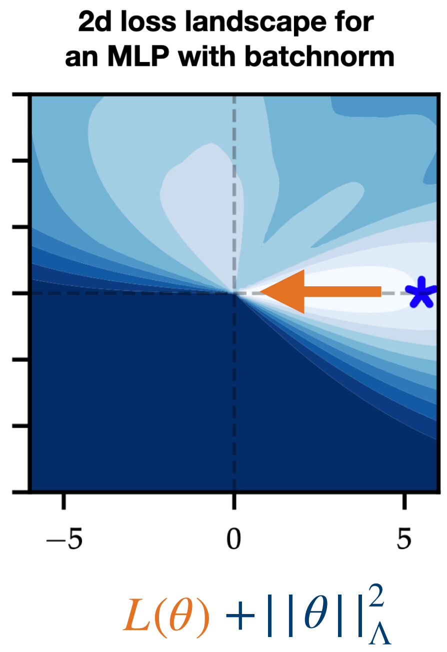
2d loss landscape for
an MLP with batchnorm



$$L(\theta) + ||\theta||_{\Lambda}^{2}$$

Issue 1: linearisation point $\tilde{\theta}$ is not a mode of the loss

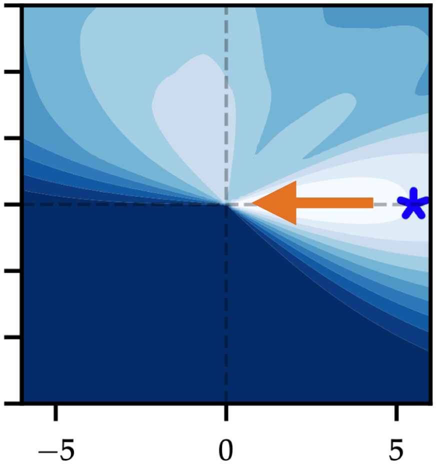
- Stochastic optimisation, early stopping or normalisation layers prevent us from identifying a mode of the loss



Issue 1: linearisation point $\tilde{\theta}$ is not a mode of the loss

- Stochastic optimisation, early stopping or normalisation layers prevent us from identifying a mode of the loss
- $\tilde{\theta}$ depends on exogenous factors & is not a mode of the linear model's loss

2d loss landscape for
an MLP with batchnorm

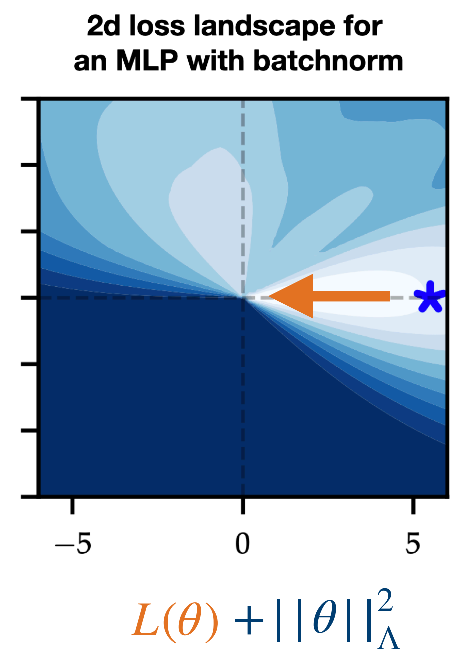


$L(\theta) + ||\theta||^2_\Lambda$

Issue 1: linearisation point $\tilde{\theta}$ is not a mode of the loss

- Stochastic optimisation, early stopping or normalisation layers prevent us from identifying a mode of the loss
- $\tilde{\theta}$ depends on exogenous factors & is not a mode of the linear model's loss

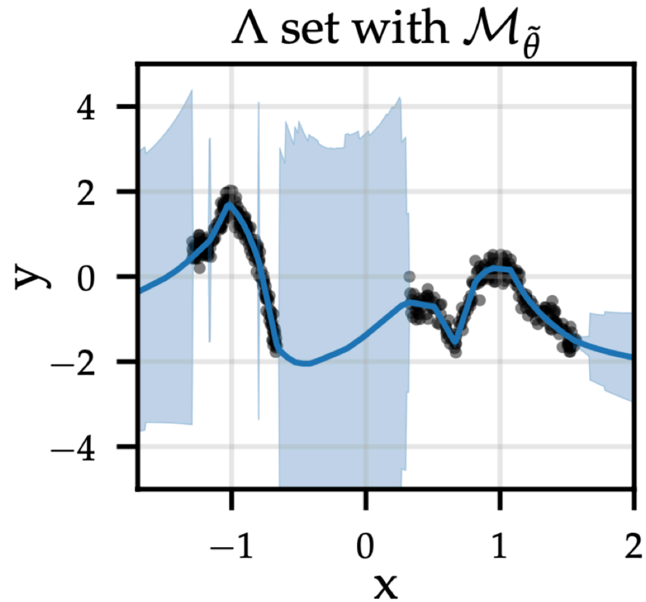
$$\mathcal{M}_{\tilde{\theta}}(\Lambda) = -\frac{1}{2}\left[\underline{\|\tilde{\theta}\|^2_\Lambda} + \log\det(\Lambda^{-1}H + I)\right] + C,$$

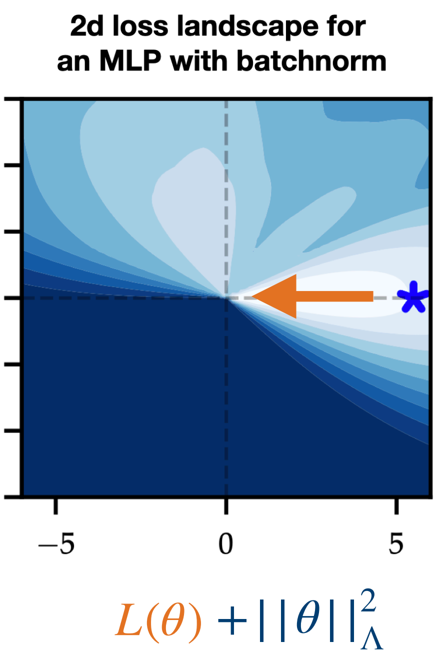


Issue 1: linearisation point $\tilde{\theta}$ is not a mode of the loss

- Stochastic optimisation, early stopping or normalisation layers prevent us from identifying a mode of the loss
- $\tilde{\theta}$ depends on exogenous factors & is not a mode of the linear model's loss

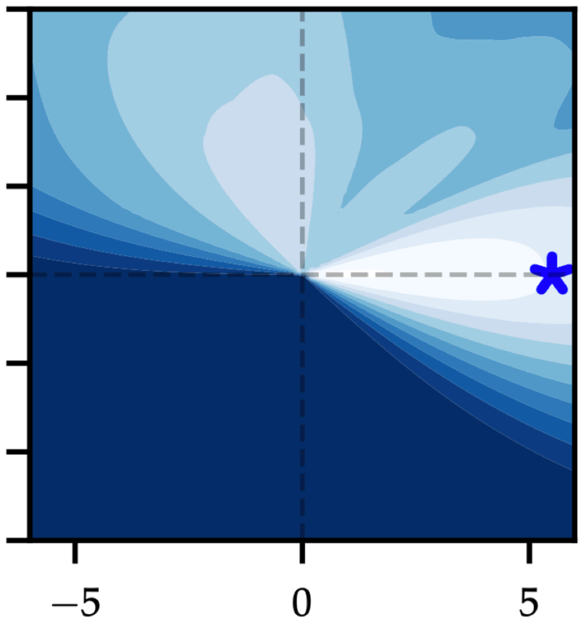
$$\mathcal{M}_{\tilde{\theta}}(\Lambda) = -\frac{1}{2}\left[\underline{\|\tilde{\theta}\|^2_\Lambda} + \log\det(\Lambda^{-1}H + I)\right] + C,$$



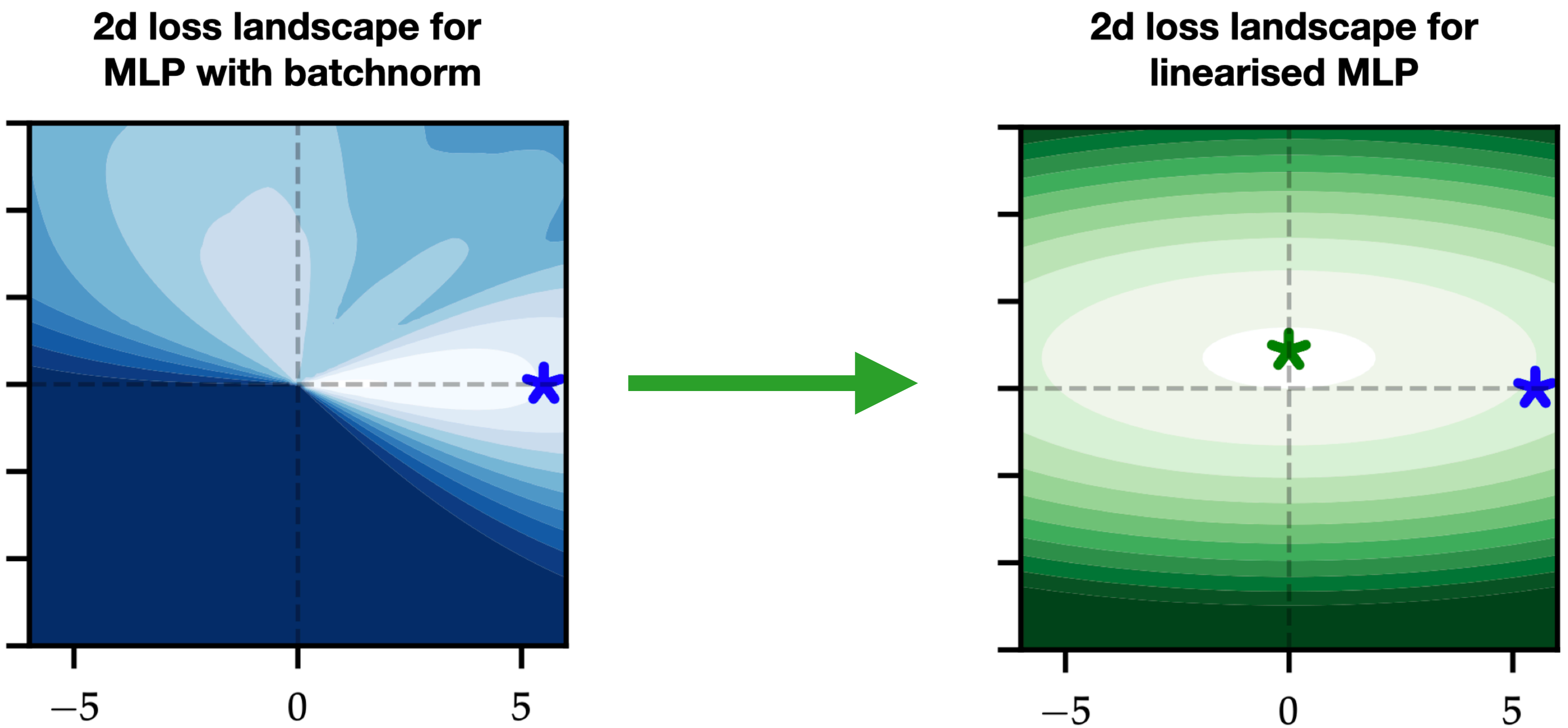


Solution 1: find mode of linear model's loss

2d loss landscape for
MLP with batchnorm



Solution 1: find mode of linear model's loss

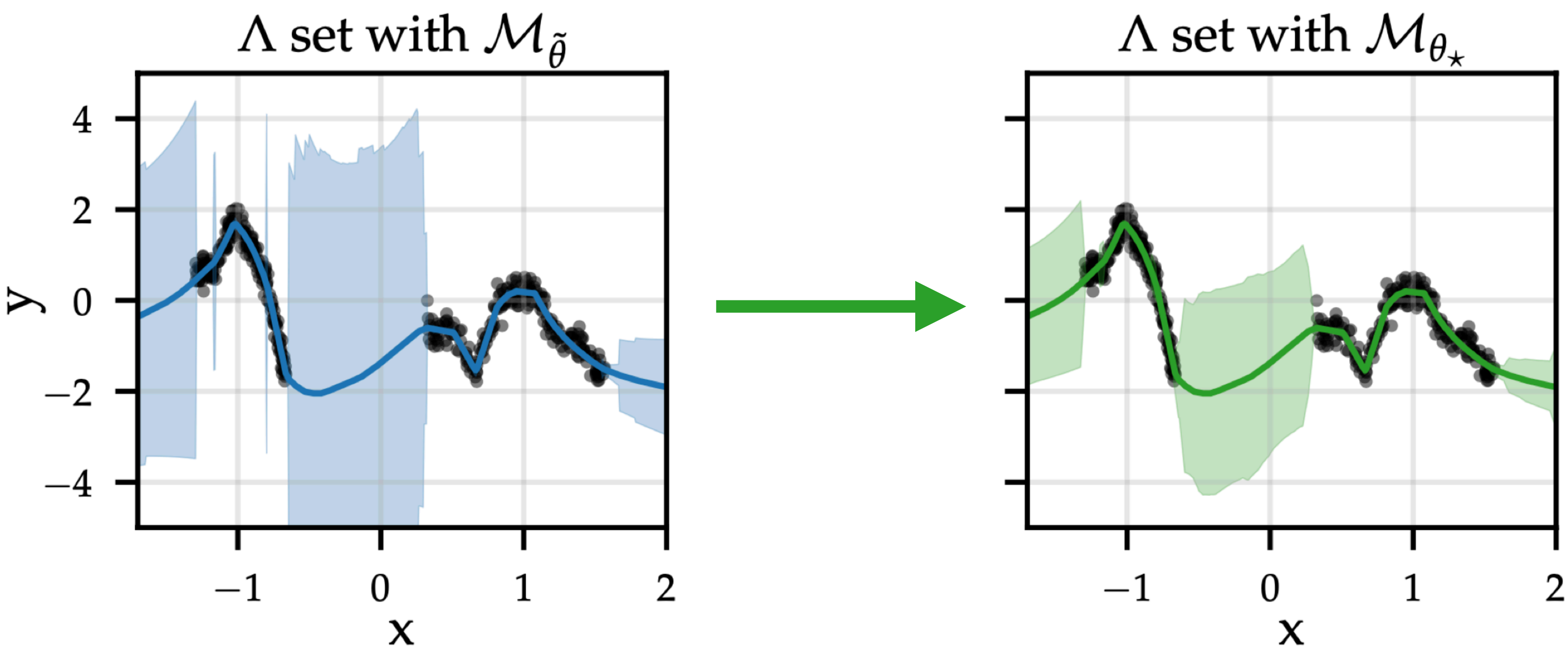


Solution 1: use this mode in the evidence expression

$$\mathcal{M}_{\theta_\star}(\Lambda) = -\frac{1}{2}\left[\|\theta_\star\|^2_\Lambda + \log\det(\Lambda^{-1}H + I)\right] + C.$$

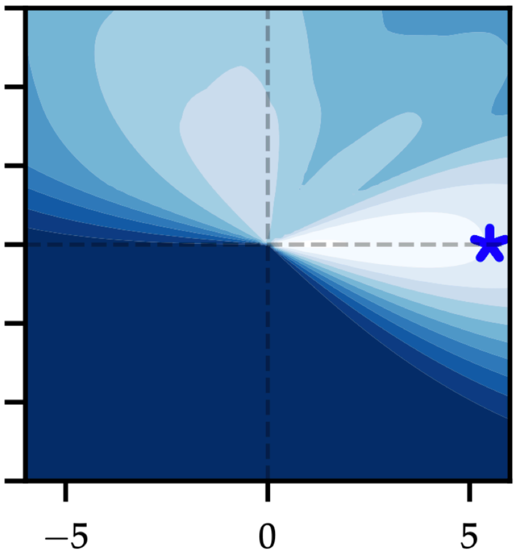
Solution 1: use this mode in the evidence expression

$$\mathcal{M}_{\theta_\star}(\Lambda) = -\frac{1}{2}\left[\|\theta_\star\|^2_\Lambda + \log\det(\Lambda^{-1}H + I)\right] + C.$$

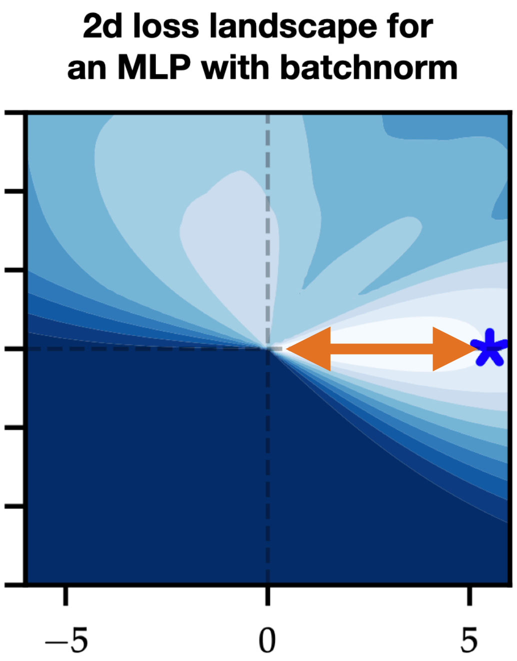


Issue 2: Dependence on scale of linearisation point k

2d loss landscape for an MLP with batchnorm

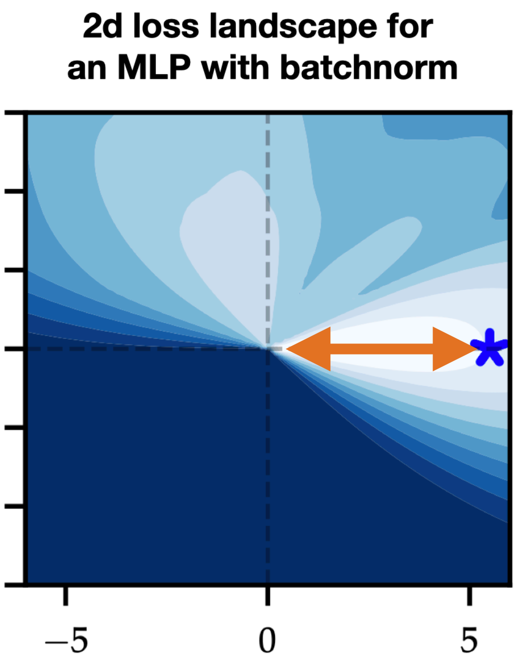


Issue 2: Dependence on scale of linearisation point k



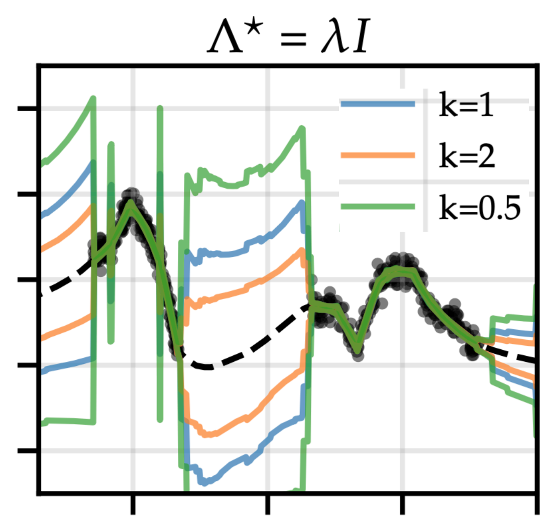
k is arbitrary and does not affect NN predictions so it should not affect the predictive variance!

Issue 2: Dependence on scale of linearisation point k



k is arbitrary and does not affect NN predictions so it should not affect the predictive variance!

However, in general, it does!



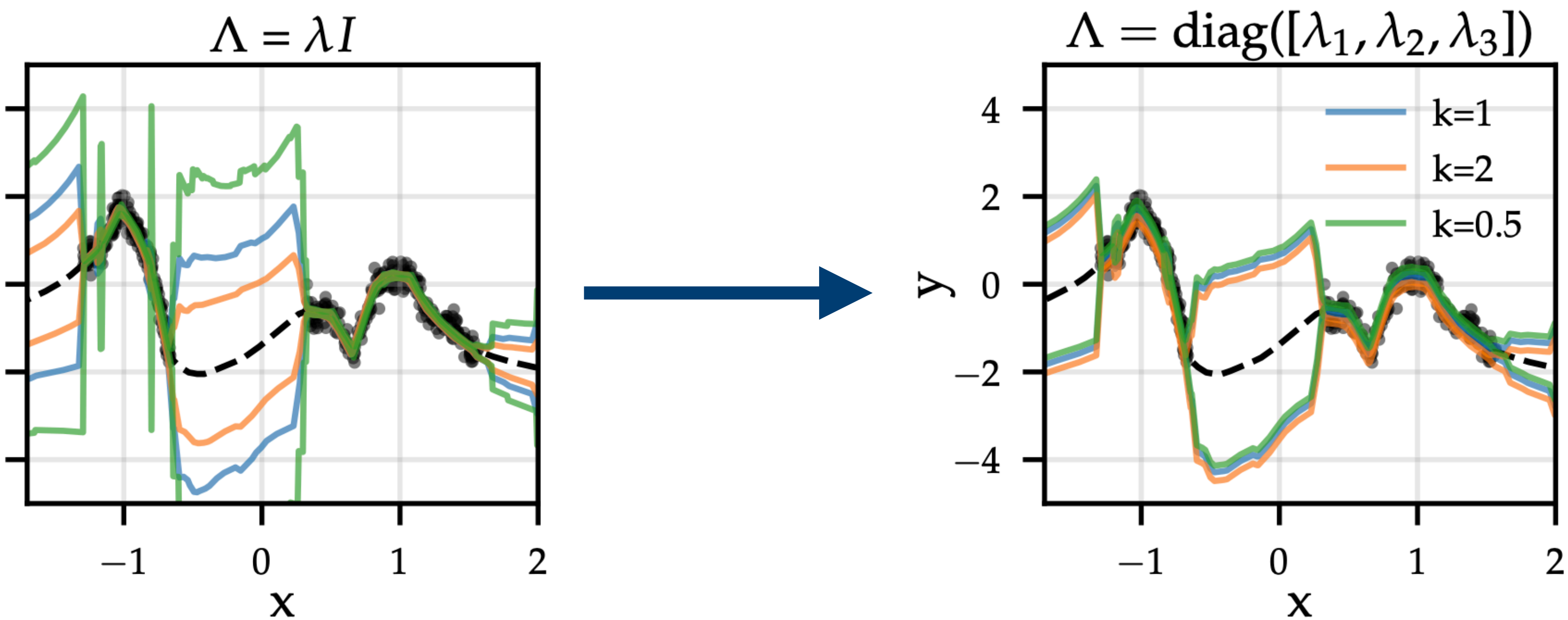
Solution 2: separately regularised normalised weight groups

Solution 2: separately regularised normalised weight groups

$$||\theta||_{\Lambda}^{2} = \lambda_0 ||\theta^{(0)}||^2 + \lambda_1 ||\theta^{(1)}||^2 + \lambda_2 ||\theta^{(2)}||^2 + \ldots$$

Solution 2: separately regularised normalised weight groups

$$||\theta||^2_\Lambda = \lambda_0||\theta^{(0)}||^2 + \lambda_1||\theta^{(1)}||^2 + \lambda_2||\theta^{(2)}||^2 + \dots$$



Closing remarks

Closing remarks

- We validate recommendations on Transformers, LeNet-style CNNs, ResNets with and without normalisation layers, and U-Net auto encoders

Closing remarks

- We validate recommendations on Transformers, LeNet-style CNNs, ResNets with and without normalisation layers, and U-Net auto encoders
- We validate at scale (21M param NN), where KFAC approximation is used for inference

Closing remarks

- We validate recommendations on Transformers, LeNet-style CNNs, ResNets with and without normalisation layers, and U-Net auto encoders
- We validate at scale (21M param NN), where KFAC approximation is used for inference
- Discuss a number of implications and interesting special cases