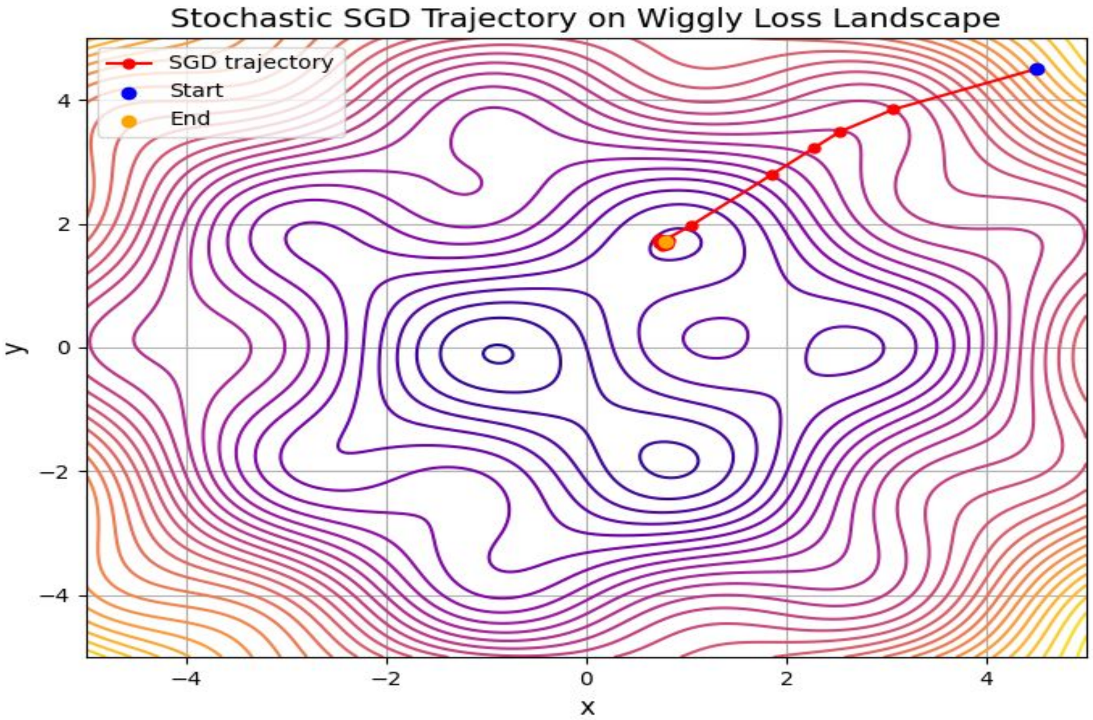
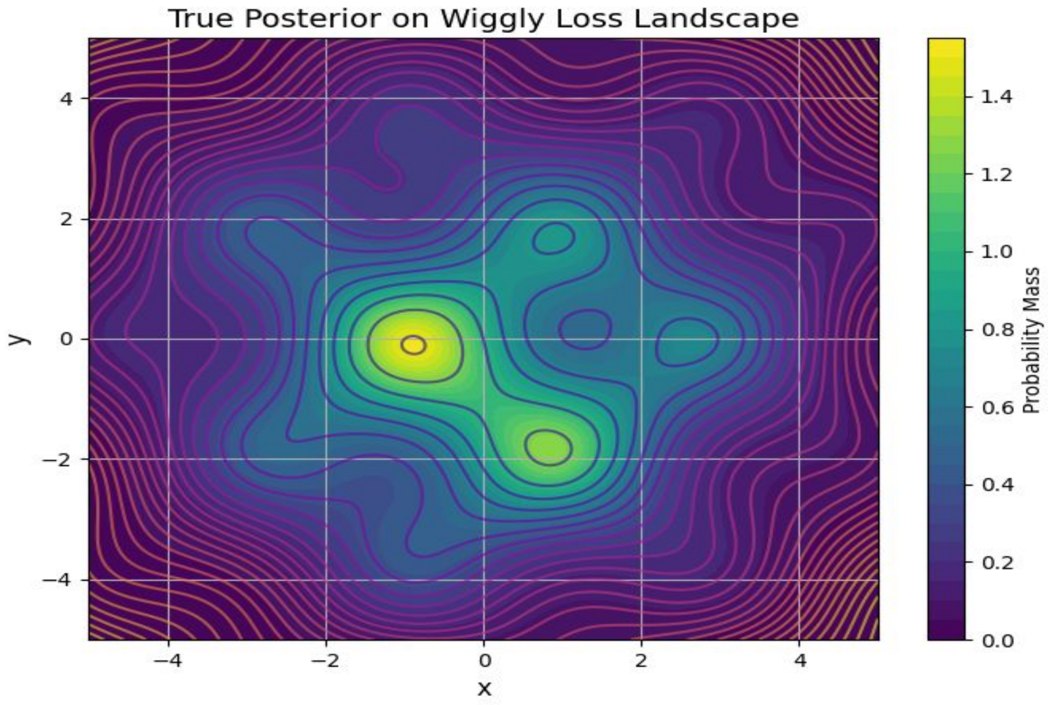
Laplace Approximations for Bayesian Deep Learning

Deep Learning



True Posterior of the Neural Network



Bayesian Deep learning

$$p(y|x,\theta) = \prod_{i=1}^{N} p(y_i|f_\theta(x)) \; : \; Likelihood$$

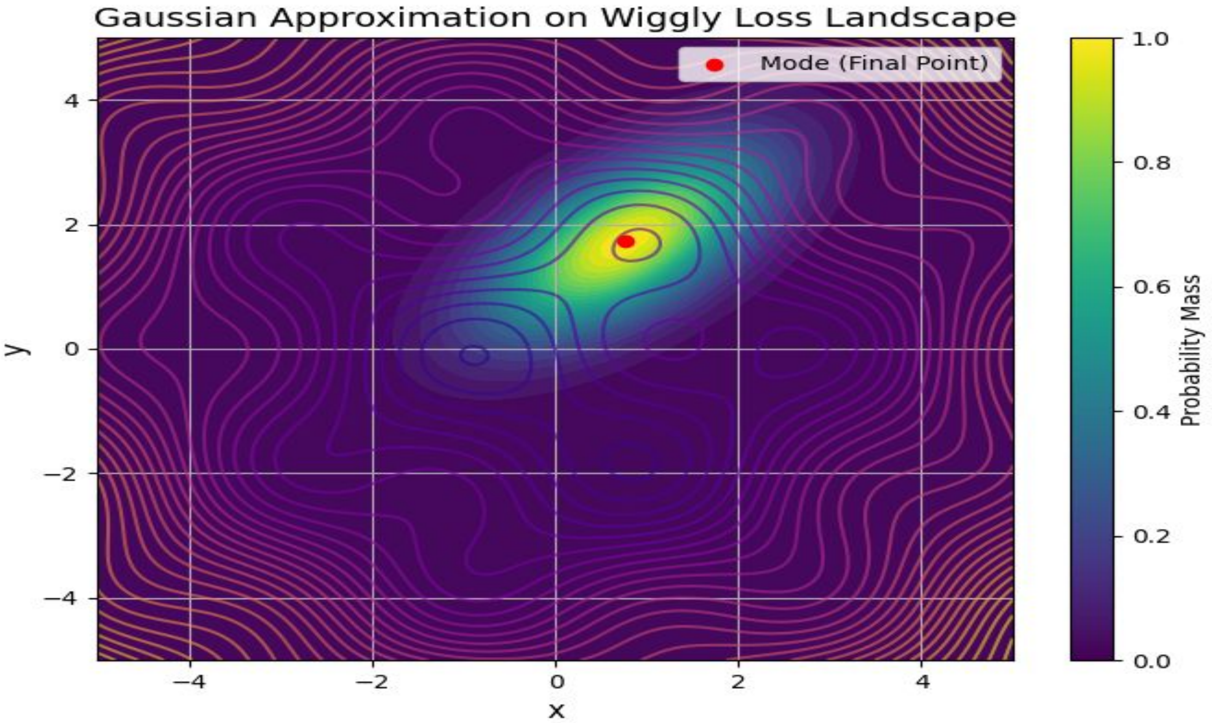
$$p(\theta) \; : \; Prior$$

Intractable

$$p(y|x) = \int_{\theta\in\Theta} p(y|x,\theta)p(\theta)d\theta \; : \; Evidence/Marginal\, Likelihood$$

$$p(\theta|x,y) \; : \; Posterior$$

Local Approximation of the True Posterior



Laplace Approximation

$$\mathcal{L}(\theta|x, y) \; = \; -\,log\, p(\mathbf{y}|f_\theta(\mathbf{x})) \; - \; log\, p_\Lambda(\theta)$$

$$\mathcal{L}(\theta|x, y) \; \approx \; \mathcal{L}(\theta_{\mathcal{MAP}}|x, y) \; + \; \frac{1}{2}(\theta - \; \theta_{\mathcal{MAP}})^T \nabla_\theta^2 \mathcal{L}\Big|_{\theta_{\mathcal{MAP}}} (\theta - \; \theta_{\mathcal{MAP}})$$

$$p(y|x, \theta) p_\Lambda(\theta) \; \approx \; \exp\left(-\,\mathcal{L}(\theta_{\mathcal{MAP}}|x, y) \;\right) \exp\left(-\frac{1}{2}(\theta - \; \theta_{\mathcal{MAP}})^T \nabla_\theta^2 \mathcal{L}\Big|_{\theta_{\mathcal{MAP}}} (\theta - \; \theta_{\mathcal{MAP}})\right)$$

Laplace Approximation

- With this approximation the evidence becomes a Gaussian Integral which is tractable

$$\int_{\theta\in\Theta} p(y|x,\theta)p(\theta)d\theta \approx \int_{\theta\in\Theta} \exp\left(-\mathcal{L}(\theta_{\mathcal{MAP}}|x,y)\right)\exp\left(-\frac{1}{2}(\theta-\theta_{\mathcal{MAP}})^T\nabla_\theta^2\mathcal{L}\Big|_{\theta_{\mathcal{MAP}}}(\theta-\theta_{\mathcal{MAP}})\right)d\theta$$

$$= \exp\left(-\mathcal{L}(\theta_{\mathcal{MAP}}|x,y)\right)\int_{\theta\in\Theta} \exp\left(-\frac{1}{2}(\theta-\theta_{\mathcal{MAP}})^T\nabla_\theta^2\mathcal{L}\Big|_{\theta_{\mathcal{MAP}}}(\theta-\theta_{\mathcal{MAP}})\right)d\theta$$

Laplace Approximation

- Under the laplace approximation the True Posterior is approximated by the following Gaussian:

$$q_{\text{LLA}}(\boldsymbol{\theta}|\mathcal{D}) = \mathcal{N}\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}_{\text{MAP}},\ (\alpha\mathbb{I} + \text{GGN}_{\boldsymbol{\theta}_{\text{MAP}}})^{-1}\right)$$

$$\text{GGN}_{\boldsymbol{\theta}_{\text{MAP}}} = \mathbf{J}_{\boldsymbol{\theta}_{\text{MAP}}}^{\top} \mathbf{H}_{\boldsymbol{\theta}_{\text{MAP}}} \mathbf{J}_{\boldsymbol{\theta}_{\text{MAP}}} \in \mathbb{R}^{P \times P}.$$

Why would we want to do this?

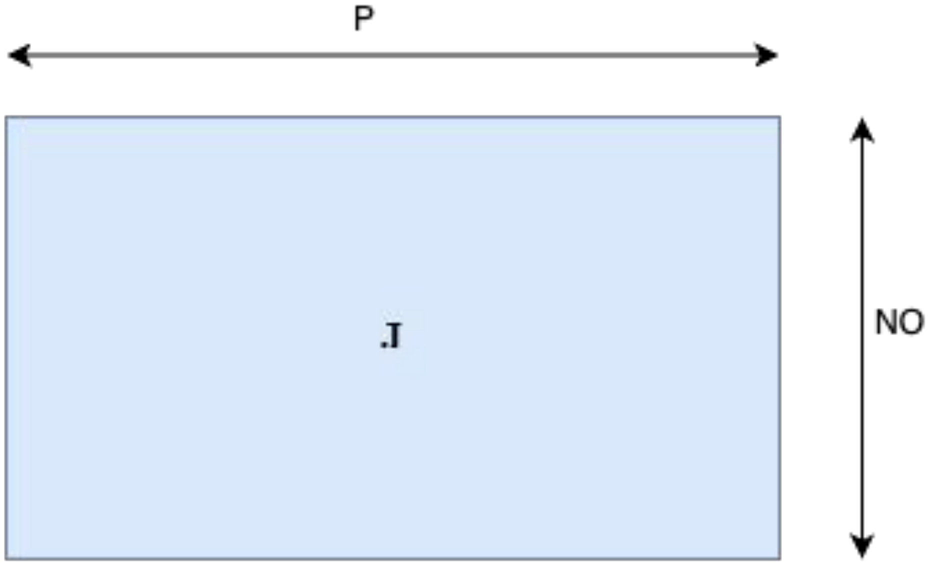
- Bayesian Model Averaging: Marginalize over the posterior for better calibrated predictions

- Principled Uncertainty Quantification

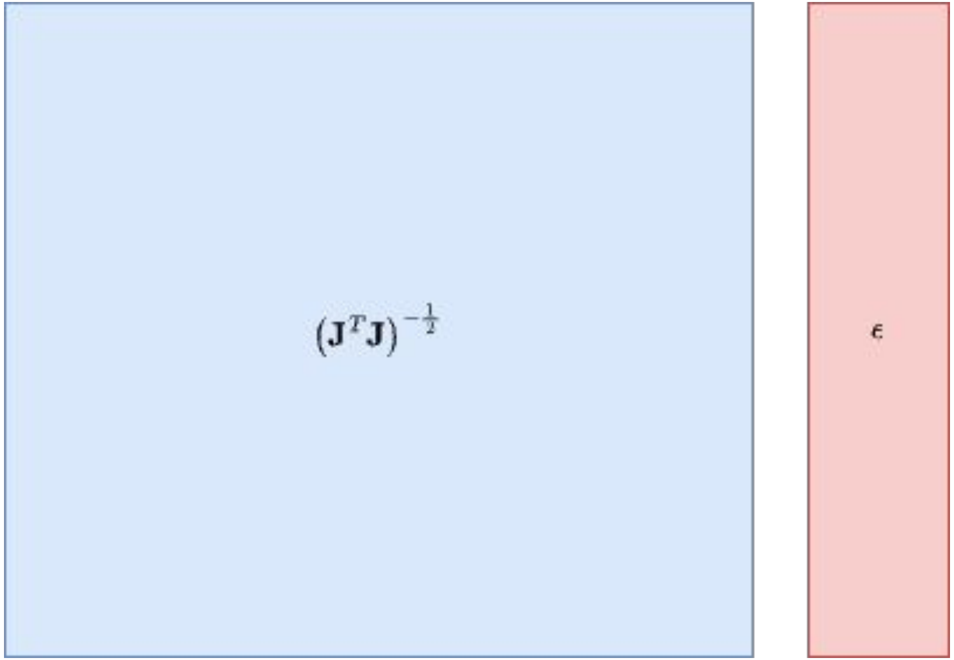
- Marginal Likelihood Optimization: Model Selection without Cross-Validation

Linear Algebra

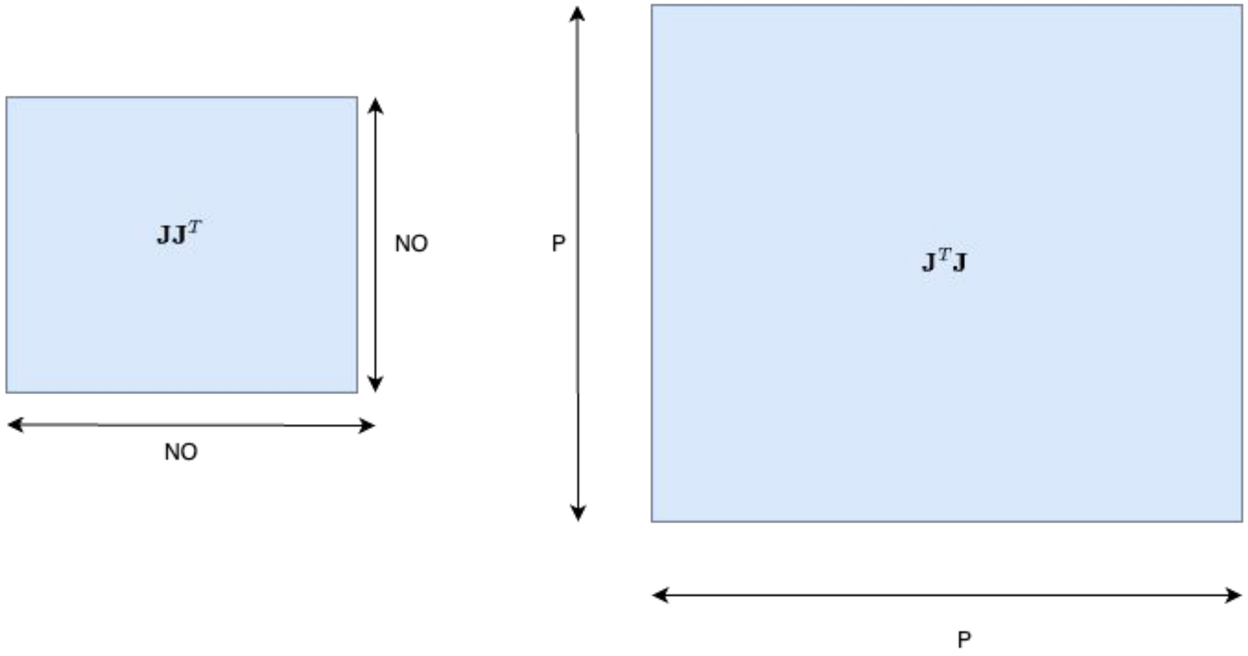
Size of the Jacobian Matrix



Sampling is hard!

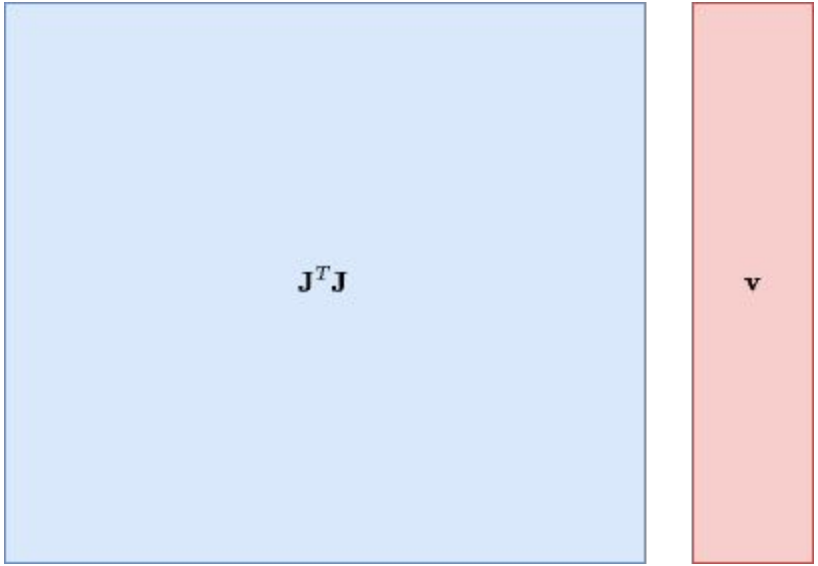


Big Matrices



- Too big to instantiate
- Direct Solvers are impractical

But Matrix-Vector Products are cheap



- Matrix Vector Products only require Jacobian Vector Products and Vector Jacobian Products(Autodiff)
- We can use Iterative Solvers!

Thank You!