Going Bayesian

through

Laplace approximation

Ivan Rodriguez

Bayesian modelling review

Given some observed data: $D = \{X_{obs}, y_{obs}\}$ and a model: $\vec{f}_{\vec{\omega}}(X) = (f^1_{\vec{\omega}}, f^2_{\vec{\omega}}, \ldots, f^d_{\vec{\omega}})$

$\vec{\omega} = (\omega_1, .., \omega_M)$

DNN with $\boldsymbol{M}$ layers and $\boldsymbol{d}$ outcomes:

We want to predict a new $\hat{y}$ given an observed feature $\hat{x}$ and the observed data D :

Predictive

$P(\hat{y}/D, \hat{x})$

Bayesian modelling review

Given some observed data: $D = \{X_{obs}, y_{obs}\}$ and a model: $\vec{f}_{\vec{\omega}}(X) = (f^1_{\vec{\omega}}, f^2_{\vec{\omega}}, \ldots, f^d_{\vec{\omega}})$

$\vec{\omega} = (\omega_1, .., \omega_M)$

DNN with $\boldsymbol{M}$ layers and $\boldsymbol{d}$ outcomes:

We want to predict a new $\hat{y}$ given an observed feature $\hat{x}$ and the observed data D :

Predictive

Posterior

$$P(\hat{y}/D, \hat{x}) = \int d\omega \; P(\hat{y}/\omega, \hat{x}) P(\omega/D)$$

Bayesian modelling review

Given some observed data: $D = \{ X_{obs}, y_{obs} \}$ and a model: $\vec{f}_{\vec{\omega}}(X) = (f^1_{\vec{\omega}}, f^2_{\vec{\omega}}, \dots, f^d_{\vec{\omega}})$

$\vec{\omega} = (\omega_1, .., \omega_M)$

DNN with $\boldsymbol{M}$ layers and $\boldsymbol{d}$ outcomes:

We want to predict a new $\hat{y}$ given an observed feature $\hat{x}$ and the observed data D :

Predictive

Posterior

$$P(\hat{y}/D, \hat{x}) = \int d\omega \; P(\hat{y}/\omega, \hat{x}) P(\omega/D)$$

- Posterior: In general very hard to compute!!
- Distribution over models vs. point-like MLE.
- Most likely models, i.e. the ones contributing more to the integral, are close to the MLE (see later).

An approximation for the posterior is needed!!

(except in simple cases, as will be clear in a moment)

Bayesian modelling review

Motivation to go Bayesian:

- Access to the probability distribution of the target, $\hat{y}$,variable: mean values, robust uncertainty estimates (Confidence Intervals), etc.

- Richer than traditional ML approach (frequentist) of Maximum Likelihood Estimation (MLE).

- Improvement of the poor calibration and overconfidence of MLE DNN (see TfL calibration training).
  (Guo et. al. ICML ’17)

- Improve of catastrophic forgetting of previously learned tasks when continuously trained on new tasks.
  (J. Kirkpatrick et. al. - Pnas ‘17; C.Nguyen et.al. ICLR ‘18)

- Allowing for automated model selection by optimally trading off data fit and model complexity.
  (F.Hutter et.al. SSCML ‘19)

Bayesian modelling review

$$P(\hat{y}/D,\hat{x}) = \int d\omega \; \boxed{P(\hat{y}/\omega,\hat{x})} \; \boxed{P(\omega/D)}$$

✔ ?

$$P(\hat{y}/\omega,\hat{x})$$

regression

$$N(\hat{y}, f_{\omega}(\hat{x}), \sigma^2) \sim e^{\frac{(f_{\omega}(\hat{x})-\hat{y})^2}{2\sigma^2}}$$

classification

$$Softmax(f_{\omega}(\hat{x}), \hat{y}=i) = e^{f_{\omega}^{i}(\hat{x})} / \sum_{j=1}^{d} e^{f_{\omega}^{j}(\hat{x})}$$

Bayesian modelling review

$$P(\hat{y}/D,\hat{x})=\int d\omega\, \boxed{P(\hat{y}/\omega,\hat{x})}\,\boxed{P(\omega/D)}$$

?

regression

$$P(\hat{y}/\omega,\hat{x}) \rightarrow N(\hat{y},f_\omega(\hat{x}),\sigma^2)\sim e^{\frac{(f_\omega(\hat{x})-\hat{y})^2}{2\sigma^2}}$$

classification

$$Softmax(f_\omega(\hat{x}),\hat{y}=i)=e^{f^i_\omega(\hat{x})}/\sum_{j=1}^{d} e^{f^j_\omega(\hat{x})}$$

In a Bayesian approach we use the **Bayes theorem** to get the posterior :

$$P(\omega/D)=\frac{P(D/\omega)P(\omega)}{P(D)}$$

Likelihood

$$P(D/\omega) \rightarrow N(y,f_\omega(X),\sigma^2)$$

$$P(D/\omega) \rightarrow Softmax(y,f_\omega(X),\sigma^2)$$

Prior

$$P(\omega)=N(\omega,\gamma^2)$$

Evidence or Normalization

$$P(D)=\int d\omega P(D,\omega)=\int d\omega P(D/\omega)P(\omega)$$

Bayesian modelling review

$$P(\hat{y}/D,\hat{x}) = \int d\omega \boxed{P(\hat{y}/\omega,\hat{x})}\boxed{P(\omega/D)}$$

✔ ?

$P(\hat{y}/\omega,\hat{x})$

regression

$$N(\hat{y}, f_\omega(\hat{x}), \sigma^2) \sim e^{\frac{(f_\omega(\hat{x})-\hat{y})^2}{2\sigma^2}}$$

classification

$$Softmax(f_\omega(\hat{x}), \hat{y}=i) = e^{f_\omega^i(\hat{x})} / \sum_{j=1}^{d} e^{f_\omega^j(\hat{x})}$$

In a Bayesian approach we use the **Bayes theorem** to get the posterior :

$$P(\omega/D) = \frac{P(D/\omega)P(\omega)}{P(D)}$$

Likelihood

$P(D/\omega)$

$N(y, f_\omega(X), \sigma^2)$

$Softmax(y, f_\omega(X), \sigma^2)$

Prior

$P(\omega) = N(\omega, \gamma^2)$

Evidence or Normalization

$$P(D) = \int d\omega P(D,\omega) = \int d\omega P(D/\omega)P(\omega)$$

In a DNN $dim(\omega)$ is too large:

intractable multi-dimensional integral !!!

Bayesian modelling review

Posterior – MLE relationship: **The log of the posterior is proportional to the MLE loss function.**

Bayesian modelling review

Posterior – MLE relationship: **The log of the posterior is proportional to the MLE loss function.**

$$\log P(\omega/D) = \log P(D/\omega) + \log P(\omega) - \log P(D)$$

e.g. for regression: $$\log P(\omega/D) = -\sum_{i=1}^{N} (f_{\omega}(X_i) - y_i)^2 - \frac{1}{\gamma^2} \sum_{l=1}^{M} \omega_l^2 - 2\sigma^2 - \log P(D)$$

Bayesian modelling review

Posterior – MLE relationship: **The log of the posterior is proportional to the MLE loss function.**

$$\log P(\omega/D) = \log P(D/\omega) + \log P(\omega) - \log P(D)$$

e.g. for regression: $$\log P(\omega/D) = \boxed{-\sum_{i=1}^{N}(f_\omega(X_i) - y_i)^2 - \frac{1}{\gamma^2}\sum_{l=1}^{M}\omega_l^2} - 2\sigma^2 - \log P(D)$$

$\stackrel{\text{def}}{=} L$ MLE + L2 reg. or MAP (Maximum a posteriori) cost function

Bayesian modelling review

Posterior – MLE relationship: **The log of the posterior is proportional to the MLE loss function.**

$$\log P(\omega/D) = \log P(D/\omega) + \log P(\omega) - \log P(D)$$

e.g. for regression: $$\log P(\omega/D) = \boxed{-\sum_{i=1}^{N}(f_{\omega}(X_i) - y_i)^2 - \frac{1}{\gamma^2}\sum_{l=1}^{M}\omega_l^2} - 2\sigma^2 - \log P(D)$$

$\stackrel{\text{def}}{=} L$ MLE + L2 reg. or MAP (Maximum a posteriori) cost function

MLE: Gradient descent

$$\omega^{\text{MAP}} = arg\,max_{\omega} \log P(\omega/D)$$

$$E(\hat{y}) = f_{\omega^{\text{MAP}}}(\hat{x})$$

Bayesian approach

$$P(\hat{y}/D, \hat{x}) = \int d\omega \;\; P(\hat{y}/\omega, \hat{x}) P(\omega/D) \simeq \sum_{\substack{\omega \\ \omega \sim P(\omega/D)}} P(\hat{y}/\omega, \hat{x})$$

$E(\hat{y})$ $Var(\hat{y})$

Bayesian modelling review

Posterior – MLE relationship: **The log of the posterior is proportional to the MLE loss function.**

$$\log P(\omega/D) = \log P(D/\omega) + \log P(\omega) - \log P(D)$$

e.g. for regression: $$\log P(\omega/D) = \boxed{-\sum_{i=1}^{N} (f_{\omega}(X_i) - y_i)^2 - \frac{1}{\gamma^2} \sum_{l=1}^{M} \omega_l^2} - 2\sigma^2 - \log P(D)$$

$\stackrel{\text{def}}{=} L$ MLE + L2 reg. or MAP (Maximum a posteriori) cost function

MLE: Gradient descent

$$\omega^{\mathrm{MAP}} = arg\,max_{\omega} \log P(\omega/D)$$

$$E(\hat{y}) = f_{\omega^{\mathrm{MAP}}}(\hat{x})$$

Bayesian approach

$$P(\hat{y}/D, \hat{x}) = \int d\omega \;\; P(\hat{y}/\omega, \hat{x}) P(\omega/D) \simeq \sum_{\substack{\omega \\ \omega \sim P(\omega/D)}} P(\hat{y}/\omega, \hat{x})$$

$E(\hat{y})$ $\quad Var(\hat{y})$

In general the $\omega's$ close to ω^{MAP} will contribute significantly to the sum.

Bayesian modelling review

Simple case where posterior can be computed analytically: Bayesian linear regression

$$P(\omega / D) = \frac{\boxed{P(D/\omega)P(\omega)}}{P(D)}$$

~ Gaussian in ω

Likelihood

$$P(D/\omega) = N(y, f_\omega(X), \sigma^2)$$

Prior

$$P(\omega) = N(\omega, \mathbf{1})$$

Normalization

$$P(D) = \int d\omega P(D, \omega)$$

Model

$$f_\omega(X) = X^T \omega$$

Bayesian modelling review

Simple case where posterior can be computed analytically: Bayesian linear regression

$$P(\omega / D) = \boxed{\frac{P(D/\omega)P(\omega)}{P(D)}}$$

~ Gaussian in ω

Likelihood

$$P(D/\omega) = N(y, f_\omega(X), \sigma^2)$$

Prior

$$P(\omega) = N(\omega, \mathbf{1})$$

Normalization

$$P(D) = \int d\omega P(D, \omega)$$

Model

$$f_\omega(X) = X^T \omega$$

$$P(\omega / D) = \underbrace{\boxed{\frac{1}{\sqrt{det(2\pi\Sigma)}}}}_{P(D)} \underbrace{\boxed{e^{-0.5*(\omega-\mu)^T \Sigma^{-1}(\omega-\mu)}}}_{P(D/\omega)P(\omega)}$$

Bayesian modelling review

Simple case where posterior can be computed analytically: Bayesian linear regression

$$P(\omega/D) = \frac{\boxed{P(D/\omega)P(\omega)}}{P(D)}$$

~ Gaussian in ω

Likelihood

$P(D/\omega) = N(y, f_\omega(X), \sigma^2)$

Prior

$P(\omega) = N(\omega, \mathbf{1})$

Normalization

$P(D) = \int d\omega P(D,\omega)$

Model

$f_\omega(X) = X^T \omega$

Issue: Unfortunately the likelihood is a softmax for classification and in the case of a DNN the model is non-linear in ω . **So, in general, there is not a close form for the posterior.**

Laplace Approximation: solve this issue by approximate $P(D/\omega)P(\omega)$ by an unnormalized Gaussian.

Laplace Approximation

$$P(\omega/D)=\frac{P(D/\omega)P(\omega)}{P(D)}=\frac{1}{P(D)}e^{L(\omega,D)}$$

$$L(\omega/D)=\log P(D/\omega)+\log P(\omega)$$

Laplace Approximation: 2nd order expansion of $L(\omega,D)$ around its maximum $\omega\sim\omega^{\mathrm{MAP}}$

$$L(\omega,D)\simeq L(\omega^{\mathrm{MAP}},D)+\frac{1}{2}(\omega-\omega^{MAP})^T\nabla_\omega^2 L(\omega,D)\big|_{\omega^{\mathrm{MAP}}}(\omega-\omega^{\mathrm{MAP}})$$

1st approximation

Laplace Approximation

$$P(\omega/D)=\frac{P(D/\omega)P(\omega)}{P(D)}=\frac{1}{P(D)}e^{L(\omega,D)}$$

$$L(\omega/D)=\log P(D/\omega)+\log P(\omega)$$

Laplace Approximation: 2^{nd} order expansion of $L(\omega,D)$ around its maximum $\omega\sim\omega^{\mathrm{MAP}}$

$$\boxed{L(\omega,D)\simeq L(\omega^{\mathrm{MAP}},D)+\frac{1}{2}(\omega-\omega^{MAP})^T\nabla^2_\omega L(\omega,D)\big|_{\omega^{\mathrm{MAP}}}(\omega-\omega^{\mathrm{MAP}})}$$

1^{st} approximation

$$P(\omega/D)=\boxed{\frac{e^{L(\omega^{\mathrm{MAP}},D)}}{P(D)}}e^{\frac{1}{2}(\omega-\omega^{MAP})^T\boxed{\nabla^2_\omega L(\omega,D)\big|_{\omega^{\mathrm{MAP}}}}(\omega-\omega^{\mathrm{MAP}})}$$

$$P(D)=e^{L(\omega^{\mathrm{MAP}},D)}\sqrt{det(2\pi\Sigma)}$$

$$\Sigma^{-1}=\nabla^2_\omega L(\omega,D)\big|_{\omega^{\mathrm{MAP}}}$$

Remember that:

$$N(\omega;\mu,\Sigma)=\frac{1}{\sqrt{det(2\pi\Sigma)}}e^{-0.5*(\omega-\mu)^T\Sigma^{-1}(\omega-\mu)}$$

Laplace Approximation

$$P(\omega/D)=\frac{P(D/\omega)P(\omega)}{P(D)}=\frac{1}{P(D)}e^{L(\omega,D)}$$

$$L(\omega/D)=\log P(D/\omega)+\log P(\omega)$$

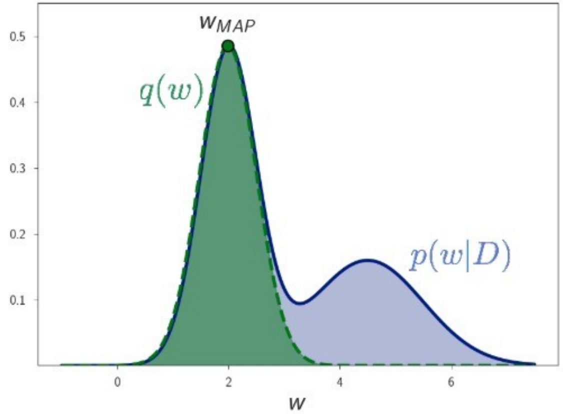
Laplace Approximation: 2nd order expansion of $L(\omega,D)$ around its maximum $\omega\sim\omega^{\mathrm{MAP}}$

$$\boxed{L(\omega,D)\simeq L(\omega^{\mathrm{MAP}},D)+\frac{1}{2}(\omega-\omega^{MAP})^T\nabla_\omega^2 L(\omega,D)\big|_{\omega^{\mathrm{MAP}}}(\omega-\omega^{\mathrm{MAP}})\qquad \textbf{1st approximation}}$$

$$P(\omega/D)=\boxed{\frac{e^{L(\omega^{\mathrm{MAP}},D)}}{P(D)}}e^{\frac{1}{2}(\omega-\omega^{MAP})^T\boxed{\nabla_\omega^2 L(\omega,D)\big|_{\omega^{\mathrm{MAP}}}}(\omega-\omega^{\mathrm{MAP}})}$$

$$P(D)=e^{L(\omega^{\mathrm{MAP}},D)}\sqrt{det(2\pi\Sigma)}$$

$$\Sigma^{-1}=\nabla_\omega^2 L(\omega,D)\big|_{\omega^{\mathrm{MAP}}}$$



Laplace Approximation

$$P(\omega/D)=\frac{P(D/\omega)P(\omega)}{P(D)}=\frac{1}{P(D)}e^{L(\omega,D)}$$

$$L(\omega/D)=\log P(D/\omega)+\log P(\omega)$$

Laplace Approximation: 2nd order expansion of $L(\omega,D)$ around its maximum $\omega\sim\omega^{\mathrm{MAP}}$

$$L(\omega,D)\simeq L(\omega^{\mathrm{MAP}},D)+\frac{1}{2}(\omega-\omega^{MAP})^T\nabla_\omega^2 L(\omega,D)\big|_{\omega^{\mathrm{MAP}}}(\omega-\omega^{\mathrm{MAP}})$$

1st approximation

In practice, to reach ω^{MAP}, a first order optimization method like gradient decent or similar will be used.

The only thing we will need to compute for the posterior will be $\Sigma^{-1}=\nabla_\omega^2 L(\omega,D)\big|_{\omega^{\mathrm{MAP}}}$ and its inverse.

Laplace Approximation

Issue 1 :

$\nabla^2_{\omega} L(\omega, D)$ It is **not semi-positive definite** so it is not a good covariance matrix.

Laplace Approximation

Issue 1 :

$\nabla^2_\omega L(\omega, D)$ It is **not semi-positive definite** so it is not a good covariance matrix.

$$\nabla^2_\omega L(\omega, D) \longrightarrow \sum_{i=1}^{N} \frac{\partial}{\partial \omega_l} \frac{\partial}{\partial \omega_m} L(\vec{f}_\omega, x_i, y_i) = \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f^\alpha_\omega}{\partial \omega_l} \frac{\partial^2 L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega \partial f^\beta_\omega} \frac{\partial f^\beta_\omega}{\partial \omega_m} + \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega} \frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m}$$

$$L(\omega, D) = \sum_{i=1}^{N} L(\omega, x_i, y_i) = \sum_{i=1}^{N} L(\vec{f}_\omega, x_i, y_i)$$

(e.g. regression $L(\omega, D) = -\sum_i (f_\omega(X_i) - y_i)^2 - \log \sum_i \omega_i^2 - 2\sigma^2 \stackrel{\text{def}}{=} \sum_i L(\omega, x_i, y_i)$)

Laplace Approximation

Issue 1 :

$\nabla^2_\omega L(\omega, D)$ It is **not semi-positive definite** so it is not a good covariance matrix.

$$\nabla^2_\omega L(\omega, D) \longrightarrow \sum_{i=1}^{N} \frac{\partial}{\partial \omega_l} \frac{\partial}{\partial \omega_m} L(\vec{f}_\omega, x_i, y_i) = \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f^\alpha_\omega}{\partial \omega_l} \frac{\partial^2 L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega \partial f^\beta_\omega} \frac{\partial f^\beta_\omega}{\partial \omega_m} + \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega} \frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m}$$

$$L(\omega, D) = \sum_{i=1}^{N} L(\omega, x_i, y_i) = \sum_{i=1}^{N} L(\vec{f}_\omega, x_i, y_i)$$

(e.g. regression $L(\omega, D) = -\sum_i (f_\omega(X_i) - y_i)^2 - \log \sum_i \omega_i^2 - 2\sigma^2 \overset{\text{def}}{=} \sum_i L(\omega, x_i, y_i)$)

$$\sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega} \frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m} \longrightarrow 0$$

(for a perfect regression or perfect classifier)

$\sim f_\omega(X_i) - y_i$ (for regression)

Laplace Approximation

Issue 1 :

$\nabla^2_\omega L(\omega, D)$ It is **not semi-positive definite** so it is not a good covariance matrix.

$$\nabla^2_\omega L(\omega, D) \longrightarrow \sum_{i=1}^{N} \frac{\partial}{\partial \omega_l} \frac{\partial}{\partial \omega_m} L(\vec{f}_\omega, x_i, y_i) = \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f^\alpha_\omega}{\partial \omega_l} \frac{\partial^2 L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega \partial f^\beta_\omega} \frac{\partial f^\beta_\omega}{\partial \omega_m} + \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega} \frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m}$$

$L(\omega, D) = \sum_{i=1}^{N} L(\omega, x_i, y_i) = \sum_{i=1}^{N} L(\vec{f}_\omega, x_i, y_i)$ (e.g. regression $L(\omega, D) = -\sum_i (f_\omega(X_i) - y_i)^2 - \log \sum_i \omega_i^2 - 2\sigma^2 \stackrel{\text{def}}{=} \sum_i L(\omega, x_i, y_i)$)

$$\sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega} \frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m} \longrightarrow 0$$

(for a perfect regression or perfect classifier)

$\sim f_\omega(X_i) - y_i$ (for regression)

$$\sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f^\alpha_\omega}{\partial \omega_l} \frac{\partial^2 L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega \partial f^\beta_\omega} \frac{\partial f^\beta_\omega}{\partial \omega_m} \stackrel{\text{def}}{=} G(\omega, D) \geq 0$$

Generalized Gaussian Newton (GGN) matrix

(See. e.g. F. Kunstner et. al. - *NeurIPS '19*; N.Schraudolph - *Neural Computation '02*)

Laplace Approximation

Issue 1 :

$\nabla^2_\omega L(\omega, D)$ It is **not semi-positive definite** so it is not a good covariance matrix.

$$\nabla^2_\omega L(\omega, D) \longrightarrow \sum_{i=1}^{N} \frac{\partial}{\partial \omega_l} \frac{\partial}{\partial \omega_m} L(\vec{f}_\omega, x_i, y_i) = \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f_\omega^\alpha}{\partial \omega_l} \frac{\partial^2 L(f_\omega, x_i, y_i)}{\partial f_\omega^\alpha \partial f_\omega^\beta} \frac{\partial f_\omega^\beta}{\partial \omega_m} + \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f_\omega^\alpha} \frac{\partial^2 f_\omega^\alpha}{\partial \omega_l \partial \omega_m}$$

$L(\omega, D) = \sum_{i=1}^{N} L(\omega, x_i, y_i) = \sum_{i=1}^{N} L(\vec{f}_\omega, x_i, y_i)$ (e.g. regression $L(\omega, D) = -\sum_i (f_\omega(X_i) - y_i)^2 - \log \sum_i \omega_i^2 - 2\sigma^2 \stackrel{\text{def}}{=} \sum_i L(\omega, x_i, y_i)$)

$$\nabla^2_\omega L(\omega, D) \simeq G(\omega, D) = \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f_\omega^\alpha}{\partial \omega_l} \frac{\partial^2 L(f_\omega, x_i, y_i)}{\partial f_\omega^\alpha \partial f_\omega^\beta} \frac{\partial f_\omega^\beta}{\partial \omega_m} \geq 0$$

2nd approximation

$$L^{(\omega, D)} \simeq L(\omega^{\mathrm{MAP}}, D) + \frac{1}{2} (\omega - \omega^{\mathrm{MAP}})^T G(\omega^{\mathrm{MAP}}, D) (\omega - \omega^{\mathrm{MAP}})$$

(See. e.g. F. Kunstner et. al. - *NeurIPS '19*; N.Schraudolph - *Neural Computation '02*)

Laplace Approximation

Issue 1 :

$\nabla^2_\omega L(\omega, D)$ It is **not semi-positive definite** so it is not a good covariance matrix.

$$\nabla^2_\omega L(\omega, D) \longrightarrow \sum_{i=1}^{N} \frac{\partial}{\partial \omega_l} \frac{\partial}{\partial \omega_m} L(\vec{f}_\omega, x_i, y_i) = \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f_\omega^\alpha}{\partial \omega_l} \frac{\partial^2 L(f_\omega, x_i, y_i)}{\partial f_\omega^\alpha \partial f_\omega^\beta} \frac{\partial f_\omega^\beta}{\partial \omega_m} + \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f_\omega^\alpha} \frac{\partial^2 f_\omega^\alpha}{\partial \omega_l \partial \omega_m}$$

Another way to support this approximation: model linearization around $\omega = \omega^{\text{MAP}}$:

(A. Immer et. al. PMLR ‘21)

$$f_\omega(x) \simeq f_{\omega^{\text{MAP}}}(x) + \sum_{l=1}^{M} \left.\frac{\partial f_\omega(x)}{\partial \omega_l}\right|_{\omega^{\text{MAP}}} (\omega_l - \omega_l^{\text{MAP}})$$

Laplace Approximation

Issue 1 :

$\nabla^2_\omega L(\omega, D)$ It is **not semi-positive definite** so it is not a good covariance matrix.

$$\nabla^2_\omega L(\omega, D) \longrightarrow \sum_{i=1}^{N} \frac{\partial}{\partial \omega_l} \frac{\partial}{\partial \omega_m} L(\vec{f}_\omega, x_i, y_i) = \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f^\alpha_\omega}{\partial \omega_l} \frac{\partial^2 L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega \partial f^\beta_\omega} \frac{\partial f^\beta_\omega}{\partial \omega_m} + \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega} \frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m}$$

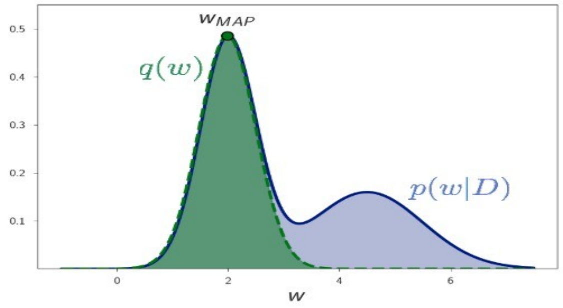
Another way to support this approximation: model linearization around $\omega = \omega^{\text{MAP}}$:

(A. Immer et. al. PMLR '21)

$$f_\omega(x) \simeq f_{\omega^{\text{MAP}}}(x) + \sum_{l=1}^{M} \left. \frac{\partial f_\omega(x)}{\partial \omega_l} \right|_{\omega^{\text{MAP}}} (\omega_l - \omega_l^{\text{MAP}})$$

$$P(\hat{y}/D, \hat{x}) = \int d\omega \; P(\hat{y}/\omega, \hat{x}) P(\omega/D) \simeq \sum_{\omega} P(\hat{y}/\omega, \hat{x})$$

main contribution from $\omega \sim \omega^{\text{MAP}}$



Laplace Approximation

Issue 1 :

$\nabla^2_\omega L(\omega, D)$ It is **not semi-positive definite** so it is not a good covariance matrix.

$$\nabla^2_\omega L(\omega,D) \longrightarrow \sum_{i=1}^{N} \frac{\partial}{\partial \omega_l}\frac{\partial}{\partial \omega_m} L(\vec{f}_\omega, x_i, y_i) = \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f^\alpha_\omega}{\partial \omega_l}\frac{\partial^2 L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega \partial f^\beta_\omega}\frac{\partial f^\beta_\omega}{\partial \omega_m} + \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega}\frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m}$$

Another way to support this approximation: model linearization around $\omega = \omega^{\mathrm{MAP}}$:

(A. Immer et. al. PMLR ‘21)

$$f_\omega(x) \simeq f_{\omega^{\mathrm{MAP}}}(x) + \sum_{l=1}^{M} \left.\frac{\partial f_\omega(x)}{\partial \omega_l}\right|_{\omega^{\mathrm{MAP}}} (\omega_l - \omega_l^{\mathrm{MAP}}) \Longrightarrow \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega} \cancelto{0}{\frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m}} \Longrightarrow \nabla^2_\omega L(\omega, D) = G(\omega, D)$$

Laplace Approximation

Issue 1 :

$\nabla^2_\omega L(\omega, D)$ It is **not semi-positive definite** so it is not a good covariance matrix.

$$\nabla^2_\omega L(\omega, D) \longrightarrow \sum_{i=1}^{N} \frac{\partial}{\partial \omega_l} \frac{\partial}{\partial \omega_m} L(\vec{f}_\omega, x_i, y_i) = \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f^\alpha_\omega}{\partial \omega_l} \frac{\partial^2 L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega \partial f^\beta_\omega} \frac{\partial f^\beta_\omega}{\partial \omega_m} + \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega} \frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m}$$

Another way to support this approximation: model linearization around $\omega = \omega^{\mathrm{MAP}}$:

(A. Immer et. al. PMLR '21)

$$f_\omega(x) \simeq f_{\omega^{\mathrm{MAP}}}(x) + \sum_{l=1}^{M} \left. \frac{\partial f_\omega(x)}{\partial \omega_l} \right|_{\omega^{\mathrm{MAP}}} (\omega_l - \omega_l^{\mathrm{MAP}}) \Longrightarrow \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega} \underbrace{\frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m}}_{\to 0} \Longrightarrow \nabla^2_\omega L(\omega, D) = G(\omega, D)$$

In particular in the linear regime:

regression

$$L(\omega, D) = L^{Laplace}(\omega, D)$$

$$L(\omega, D) = L(\omega^{\mathrm{MAP}}, D) + \frac{1}{2}(\omega - \omega^{MAP})^T G(\omega^{\mathrm{MAP}}, D)(\omega - \omega^{\mathrm{MAP}}) + \underbrace{O(\partial^3 L_i(f_\omega) / \partial f^\alpha_\omega \partial f^\beta_\omega \partial f^\gamma_\omega + \ldots)}_{\to 0}$$

($\partial^3 L_i(f_\omega) / \partial f^\alpha_\omega \partial f^\beta_\omega \partial f^\gamma_\omega$ and higher vanishes).

Laplace Approximation

Issue 1 :

$\nabla^2_\omega L(\omega, D)$ It is **not semi-positive definite** so it is not a good covariance matrix.

$$\nabla^2_\omega L(\omega, D) \longrightarrow \sum_{i=1}^{N} \frac{\partial}{\partial \omega_l} \frac{\partial}{\partial \omega_m} L(\vec{f}_\omega, x_i, y_i) = \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f^\alpha_\omega}{\partial \omega_l} \frac{\partial^2 L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega \partial f^\beta_\omega} \frac{\partial f^\beta_\omega}{\partial \omega_m} + \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega} \frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m}$$

Another way to support this approximation: model linearization around $\omega = \omega^{\text{MAP}}$:

(A. Immer et. al. PMLR '21)

$$f_\omega(x) \simeq f_{\omega^{\text{MAP}}}(x) + \sum_{l=1}^{M} \left.\frac{\partial f_\omega(x)}{\partial \omega_l}\right|_{\omega^{\text{MAP}}} (\omega_l - \omega_l^{\text{MAP}}) \Longrightarrow \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_\omega, x_i, y_i)}{\partial f^\alpha_\omega} \underbrace{\cancel{\frac{\partial^2 f^\alpha_\omega}{\partial \omega_l \partial \omega_m}}}_{0} \Longrightarrow \nabla^2_\omega L(\omega, D) = G(\omega, D)$$

In particular in the linear regime:

regression

$L(\omega, D) = L^{Laplace}(\omega, D)$

($\partial^3 L_i(f_\omega)/\partial f^\alpha_\omega \partial f^\beta_\omega \partial f^\gamma_\omega$ and higher vanishes).

classification

$L(\omega, D) \simeq L^{Laplace}(\omega, D)$

Laplace Approximation

Issue 1 :

$\nabla^2_{\omega} L(\omega, D)$ It is **not semi-positive definite** so it is not a good covariance matrix.

$$\nabla^2_{\omega} L(\omega, D) \longrightarrow \sum_{i=1}^{N} \frac{\partial}{\partial \omega_l} \frac{\partial}{\partial \omega_m} L(\vec{f}_{\omega}, x_i, y_i) = \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial f^{\alpha}_{\omega}}{\partial \omega_l} \frac{\partial^2 L(f_{\omega}, x_i, y_i)}{\partial f^{\alpha}_{\omega} \partial f^{\beta}_{\omega}} \frac{\partial f^{\beta}_{\omega}}{\partial \omega_m} + \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_{\omega}, x_i, y_i)}{\partial f^{\alpha}_{\omega}} \frac{\partial^2 f^{\alpha}_{\omega}}{\partial \omega_l \partial \omega_m}$$

Another way to support this approximation: model linearization around $\omega = \omega^{\mathrm{MAP}}$:

(A. Immer et. al. PMLR '21)

$$f_{\omega}(x) \simeq f_{\omega^{\mathrm{MAP}}}(x) + \sum_{l=1}^{M} \left. \frac{\partial f_{\omega}(x)}{\partial \omega_l} \right|_{\omega^{\mathrm{MAP}}} (\omega_l - \omega_l^{\mathrm{MAP}}) \Longrightarrow \sum_{i=1}^{N} \sum_{\alpha=1,\beta=1}^{d} \frac{\partial L(f_{\omega}, x_i, y_i)}{\partial f^{\alpha}_{\omega}} \underbrace{\frac{\partial^2 f^{\alpha}_{\omega}}{\partial \omega_l \partial \omega_m}}_{\to 0} \Longrightarrow \nabla^2_{\omega} L(\omega, D) = G(\omega, D)$$

In particular in the linear regime:

$$\boxed{P(\omega/D) \simeq \frac{1}{P(D)} e^{L^{Laplace}(\omega, D, G(\omega, D))} \quad \text{Is a very good approximation !!}}$$

Laplace Approximation

Issue 2 :

For large DNN $G^{-1}(\omega, D)$, is practically impossible to compute ($O(|\omega|^3)$) .

Laplace Approximation

Issue 2 :

For large DNN $G^{-1}(\omega, D)$, is practically impossible to compute ($O(|\omega|^3)$) .

It can be shown that $G(\omega, D)$, for many cost functions of interest (including regression and classification), is equal to the empirical Fisher matrix: (F. Kunstner et. al. - *NeurIPS '19)*

$$F(\vec{\omega}, D) = \sum_{i=1}^{N} \nabla_{\vec{\omega}} \log p(y_i / x_i, \vec{\omega}) \nabla_{\vec{\omega}} \log p(y_i / x_i, \vec{\omega})^T \qquad \vec{\omega} = (\omega_1, .., \omega_M)$$

Laplace Approximation

Issue 2 :

For large DNN $G^{-1}(\omega, D)$, is practically impossible to compute ($O(|\omega|^3)$) .

It can be shown that $G(\omega, D)$, for many cost functions of interest (including regression and classification), is equal to the empirical Fisher matrix: (F. Kunstner et. al. - *NeurIPS '19)*

$$F(\vec{\omega}, D) = \sum_{i=1}^{N} \nabla_{\vec{\omega}} \log p(y_i / x_i, \vec{\omega}) \nabla_{\vec{\omega}} \log p(y_i / x_i, \vec{\omega})^T \qquad \vec{\omega} = (\omega_1, .., \omega_M)$$

Notation:

$$\log p(y_i / x_i, \vec{\omega}) = L^{MLE}(x_i, y_i) = L_i$$

$$(\omega_i)_{\gamma,\mu} \rightarrow (\omega_i)_\alpha$$

$i =$ DNN layer number

$\gamma, \mu = 1, .., N \quad \alpha = 1, .., N^2$

$$\Longrightarrow \quad F_{ij;\alpha\beta}(\vec{\omega}, D) = \sum_{n=1}^{N} (\nabla_{\omega_{i,\alpha}} L_n)(\nabla_{\omega_{j,\beta}} L_n)^T$$

Laplace Approximation

Issue 2 :

For large DNN $G^{-1}(\omega, D)$, is practically impossible to compute ($O(|\omega|^3)$) .

It can be shown that $G(\omega, D)$, for many cost functions of interest (including regression and classification), is equal to the empirical Fisher matrix: (F. Kunstner et. al. - *NeurIPS '19)*

$$F(\vec{\omega}, D) = \sum_{i=1}^{N} \nabla_{\vec{\omega}} \log p(y_i / x_i, \vec{\omega}) \nabla_{\vec{\omega}} \log p(y_i / x_i, \vec{\omega})^T \qquad \vec{\omega} = (\omega_1, .., \omega_M)$$

Notation:

$$F_{ij;\alpha\beta}(\vec{\omega}, D) = \sum_{n=1}^{N} (\nabla_{\omega_{i,\alpha}} L_n)(\nabla_{\omega_{j,\beta}} L_n)^T$$

$$L_n = L^{MLE}(x_n, y_n)$$

$$F(\omega, D) = \begin{pmatrix} \sum_n \nabla_{\omega_1} L_n \nabla_{\omega_1} L_n^T & \sum_n \nabla_{\omega_1} L_n \nabla_{\omega_2} L_n^T & \dots & \sum_n \nabla_{\omega_1} L_n \nabla_{\omega_N} L_n^T \\ \sum_n \nabla_{\omega_2} L_n \nabla_{\omega_1} L_n^T & \sum_n \nabla_{\omega_2} L_n \nabla_{\omega_2} L_n^T & \dots & \sum_n \nabla_{\omega_2} L_n \nabla_{\omega_N} L_n^T \\ \dots & \dots & \dots & \dots \\ \boxed{\sum_n \nabla_{\omega_N} L_n \nabla_{\omega_1} L_n^T} & \sum_n \nabla_{\omega_N} L_n \nabla_{\omega_2} L_n^T & \dots & \sum_n \nabla_{\omega_N} L_n \nabla_{\omega_N} L_n^T \end{pmatrix}$$

$$F_{N,1} \sim O(|\omega_N| \times |\omega_1|) \text{ matrix}$$

Laplace Approximation

Issue 2 :

For large DNN $G^{-1}(\omega, D)$, is practically impossible to compute ($O(|\omega|^3)$) .

It can be shown that $G(\omega, D)$, for many cost functions of interest (including regression and classification), is equal to the empirical Fisher matrix: (F. Kunstner et. al. - *NeurIPS '19)*

$$F(\vec{\omega}, D) = \sum_{i=1}^{N} \nabla_{\vec{\omega}} \log p(y_i / x_i, \vec{\omega}) \nabla_{\vec{\omega}} \log p(y_i / x_i, \vec{\omega})^T \qquad \vec{\omega} = (\omega_1, .., \omega_M)$$

Notation:

$$F_{ij;\alpha\beta}(\vec{\omega}, D) = \sum_{n=1}^{N} (\nabla_{\omega_{i,\alpha}} L_n)(\nabla_{\omega_{j,\beta}} L_n)^T \longrightarrow F(\omega, D) = \begin{pmatrix} \sum_n \nabla_{\omega_1} L_n \nabla_{\omega_1} L_n^T & \sum_n \nabla_{\omega_1} L_n \nabla_{\omega_2} L_n^T & \dots & \sum_n \nabla_{\omega_1} L_n \nabla_{\omega_N} L_n^T \\ \sum_n \nabla_{\omega_2} L_n \nabla_{\omega_1} L_n^T & \sum_n \nabla_{\omega_2} L_n \nabla_{\omega_2} L_n^T & \dots & \sum_n \nabla_{\omega_2} L_n \nabla_{\omega_N} L_n^T \\ \dots & \dots & \dots & \dots \\ \boxed{\sum_n \nabla_{\omega_N} L_n \nabla_{\omega_1} L_n^T} & \sum_n \nabla_{\omega_N} L_n \nabla_{\omega_2} L_n^T & \dots & \sum_n \nabla_{\omega_N} L_n \nabla_{\omega_N} L_n^T \end{pmatrix}$$

$$L_n = L^{MLE}(x_n, y_n)$$

$$F_{N,1} \sim O(|\omega_N| \times |\omega_1|) \text{ matrix}$$

Expectation value hard to compute!!!

Laplace Approximation

Issue 2 :

For large DNN $G^{-1}(\omega, D)$, is practically impossible to compute ($O(|\omega|^3)$) .

Fisher matrix approximations (J. Martens et. al. - *ICML ‘15,* H. Ritter et.al. - *ICLR ‘18,* Daxberger et. al. - NeurIPS ‘21*)*

Diagonal approximation: $F_{ij;\alpha\beta}(\vec{\omega}, D) \simeq F_{ij;\alpha\beta}(\vec{\omega}, D)\,\delta_{i,j}\,\delta_{\alpha,\beta}$

$$F(\omega, D) = \begin{pmatrix} Diag\left(\sum_n \nabla_{\omega_1} L_n \nabla_{\omega_1} L_n^T\right) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & Diag\left(\sum_n \nabla_{\omega_2} L_n \nabla_{\omega_2} L_n^T\right) & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & Diag\left(\sum_n \nabla_{\omega_N} L_n \nabla_{\omega_N} L_n^T\right) \end{pmatrix}$$

Laplace Approximation

Issue 2 :

For large DNN $G^{-1}(\omega, D)$, is practically impossible to compute ($O(|\omega|^3)$) .

Fisher matrix approximations (J. Martens et. al. - *ICML '15,* H. Ritter et.al. - *ICLR '18,* Daxberger et. al. - NeurIPS '21)

Diagonal approximation: $F_{ij;\alpha\beta}(\vec{\omega}, D) \simeq F_{ij;\alpha\beta}(\vec{\omega}, D)\,\delta_{i,j}\,\delta_{\alpha,\beta}$

$$F(\omega, D) = \begin{pmatrix} Diag\left(\sum_n \nabla_{\omega_1} L_n \nabla_{\omega_1} L_n^T\right) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & Diag\left(\sum_n \nabla_{\omega_2} L_n \nabla_{\omega_2} L_n^T\right) & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & Diag\left(\sum_n \nabla_{\omega_N} L_n \nabla_{\omega_N} L_n^T\right) \end{pmatrix}$$

Pros:

Inverse is trivial

Expectations are easy to Compute

Cons:

Rude approximation

Laplace Approximation

Issue 2 :

For large DNN $G^{-1}(\omega, D)$, is practically impossible to compute ($O(|\omega|^3)$) .

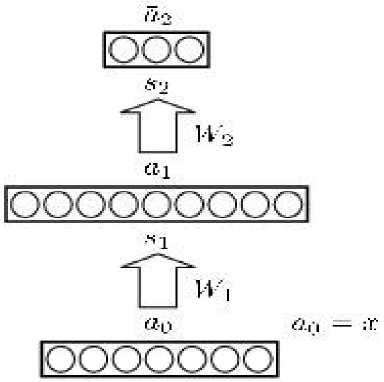
Fisher matrix approximations (J. Martens et. al. - *ICML '15,* H. Ritter et.al. - *ICLR '18,* Daxberger et. al. - NeurIPS '21*)*

Kronecker-factored approximate curvature (KFAC):

$$F_{ij} = \sum_n \nabla_{\omega_i} L_n \nabla_{\omega_j} L_n^T = E[a_{i-1} a_{j-1}^T \otimes g_i g_j^T]$$

$$g_i = \partial L / \partial s_i$$

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix},$$



Laplace Approximation

Issue 2 :

For large DNN $G^{-1}(\omega, D)$, is practically impossible to compute ($O(|\omega|^3)$) .

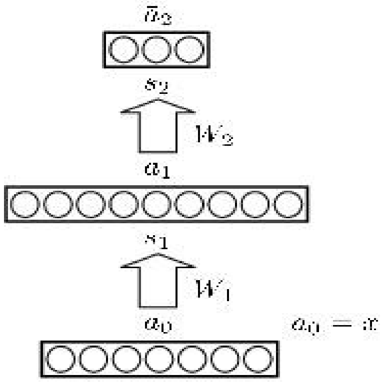
Fisher matrix approximations (J. Martens et. al. - *ICML '15,* H. Ritter et.al. - *ICLR '18,* Daxberger et. al. - NeurIPS '21*)*

Kronecker-factored approximate curvature (KFAC):

$$\boxed{F_{ij} = \sum_n \nabla_{\omega_i} L_n \nabla_{\omega_j} L_n^T = E[a_{i-1} a_{j-1}^T \otimes g_i g_j^T] \simeq E[a_{i-1} a_{j-1}^T] \otimes E[g_i g_j^T]}$$

$$g_i = \partial L / \partial s_i$$

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix},$$



Laplace Approximation

Issue 2 :

For large DNN $G^{-1}(\omega, D)$, is practically impossible to compute ($O(|\omega|^3)$) .

Fisher matrix approximations (J. Martens et. al. - *ICML '15,* H. Ritter et.al. - *ICLR '18,* Daxberger et. al. - NeurIPS '21*)*

Kronecker-factored approximate curvature (KFAC):

$$F_{ij} = \sum_n \nabla_{\omega_i} L_n \nabla_{\omega_j} L_n^T = E[a_{i-1} a_{j-1}^T \otimes g_i g_j^T] \simeq E[a_{i-1} a_{j-1}^T] \otimes E[g_i g_j^T]$$

Pros:

Much better approx. than diagonal

Expectation values easier to compute

Cons:

Still hard to compute the inverse of F

Laplace Approximation

Issue 2 :

For large DNN $G^{-1}(\omega, D)$, is practically impossible to compute ($O(|\omega|^3)$) .

Fisher matrix approximations (J. Martens et. al. - *ICML '15*, H. Ritter et.al. - *ICLR '18*, Daxberger et. al. - NeurIPS '21)

Kronecker-factored approximate curvature (KFAC):

$$\boxed{F_{ij} = \sum_n \nabla_{\omega_i} L_n \nabla_{\omega_j} L_n^T = E[a_{i-1} a_{j-1}^T \otimes g_i g_j^T] \simeq E[a_{i-1} a_{j-1}^T] \otimes E[g_i g_j^T]}$$

Pros:

Much better approx. than diagonal

Expectation values easier to compute

Inverse much cheaper

Diagonal KFAC

$$F(\omega, D) = \begin{pmatrix} F_{11}^{KFAC} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & F_{22}^{KFAC} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & F_{NN}^{KFAC} \end{pmatrix}$$

Tri-diagonal KFAC

$$F(\omega, D) = \begin{pmatrix} F_{11}^{KFAC} & F_{12}^{KFAC} & \dots & \mathbf{0} \\ F_{21}^{KFAC} & F_{22}^{KFAC} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & F_{NN}^{KFAC} \end{pmatrix}$$

Laplace Approximation

Issue 3 :

Predictive integral approximation:

$$P(\hat{y}/D,\hat{x}) \simeq \int d\omega \; P(\hat{y}/\omega,\hat{x}) P^{\text{Laplace}}(\omega/D)$$

$P(\hat{y}/\omega,\hat{x})$ → $N(\hat{y}, f_\omega(\hat{x}), \sigma^2)$ regression

$P(\hat{y}/\omega,\hat{x})$ → $Softmax(\hat{y}, f_\omega(\hat{x}), \sigma^2)$ classification

As the model, f_ω , is in general highly non-linear in ω the integral cannot be computed in a close from.

Laplace Approximation

Issue 3 :

Predictive integral approximation:

$$P(\hat{y}/D,\hat{x}) \simeq \int d\omega \; P(\hat{y}/\omega,\hat{x}) P^{\text{Laplace}}(\omega/D)$$

$P(\hat{y}/\omega,\hat{x})$ → $N(\hat{y}, f_{\omega}(\hat{x}), \sigma^2)$ regression

$P(\hat{y}/\omega,\hat{x})$ → $Softmax(\hat{y}, f_{\omega}(\hat{x}), \sigma^2)$ classification

As the model, f_{ω} , is in general highly non-linear in ω the integral cannot be computed in a close from.

Montecarlo: poor results mainly because of GGN approximation.

(GGN approx. good around linear regime but not guarantee to work beyond it!)

$$P(\hat{y}/D,\hat{x}) \simeq \frac{1}{S}\sum_{i=1}^{S} P(\hat{y}/\omega_i,\hat{x})$$

$$\omega_i \sim P^{\text{Laplace}}(\omega/D)$$

Laplace Approximation

Issue 3 :

Predictive integral approximation:

$$P(\hat{y}/D,\hat{x}) \simeq \int d\omega \ P(\hat{y}/\omega,\hat{x}) P^{\text{Laplace}}(\omega/D)$$

$P(\hat{y}/\omega,\hat{x})$ → $N(\hat{y}, f_\omega(\hat{x}), \sigma^2)$ regression

$P(\hat{y}/\omega,\hat{x})$ → $Softmax(\hat{y}, f_\omega(\hat{x}), \sigma^2)$ classification

As the model, f_ω , is in general highly non-linear in ω the integral cannot be computed in a close from.

Approximation: model linearization $f_\omega(x) \simeq f_\omega^{\text{MAP}}(x) + \sum_{l=1}^{M} \left.\frac{\partial f_\omega(x)}{\partial \omega_l}\right|_{\omega^{\text{MAP}}} (\omega_l - \omega_l^{\text{MAP}})$

- Regression case: $P(\hat{y}/\omega,\hat{x})$ becomes a Gaussian in ω and $P(\hat{y}/D,\hat{x})$ becomes a Normal distribution.

- Classification case: after 'probit' approximation $P(\hat{y}/D,\hat{x})$ becomes a Categorical distribution.
(D. J. Spiegelhalter et.al. - *Networks '90;* C.M. Bishop - *Springer '06*)

Laplace Approximation

Sub-network approximation: Replace the Bayesian network by a Bayesian sub-network 'S':
(E. Daxberger et.al - *ICML '21,* E. Daxberger et.al – *PMLR '21*)

$$P_S^{\text{Laplace}}(\omega / D) \simeq P^{\text{Laplace}}(\omega_S / D) \prod_r \delta(\omega_r - \omega_r^{\text{MAP}}) = N(\omega_S, \omega_S^{\text{MAP}}, G_S^{-1}) \prod_r \delta(\omega_r - \omega_r^{\text{MAP}})$$

probabilistic

deterministic

$$|G_s| \ll |G|$$

Laplace Approximation

Sub-network approximation: Replace the Bayesian network by a Bayesian sub-network ‘S’:
(E. Daxberger et.al - *ICML ‘21,* E. Daxberger et.al – *PMLR ’21*)

$$P_S^{\text{Laplace}}(\omega/D) \simeq P^{\text{Laplace}}(\omega_S/D)\prod_r \delta(\omega_r - \omega_r^{\text{MAP}}) = N(\omega_S, \omega_S^{\text{MAP}}, G_S^{-1})\prod_r \delta(\omega_r - \omega_r^{\text{MAP}})$$

probabilistic deterministic

$$|G_s| \ll |G|$$

Sub-network choice:

Last layer: this is an approximation that works very well as we will see later in the exercise.

Laplace Approximation

Sub-network approximation: Replace the Bayesian network by a Bayesian sub-network ‘S’:
(E. Daxberger et.al - *ICML ‘21,* E. Daxberger et.al – *PMLR ’21*)

$$P_S^{\text{Laplace}}(\omega/D) \simeq P^{\text{Laplace}}(\omega_S/D) \prod_r \delta(\omega_r - \omega_r^{\text{MAP}}) = N(\omega_S, \omega_S^{\text{MAP}}, G_S^{-1}) \prod_r \delta(\omega_r - \omega_r^{\text{MAP}})$$

probabilistic deterministic

$$|G_s| \ll |G|$$

Sub-network choice:

Last layer: this is an approximation that works very well as we will see later in the exercise.

Optimal sub-network: Select the percentage of weights of the sub-network and ‘minimize distance’ between $P_S^{\text{Laplace}}(\omega/D)$ and $P^{\text{Laplace}}(\omega/D)$.

Important: G_s^{-1} is fully computed but G^{-1} is diagonal approximated.

Notebooks

SUMMARY:

- **Laplace approximation**: $P(\omega / D) = \frac{1}{P(D)} e^{L^{Laplace}(\omega, D)}$ with $L^{\mathrm{Laplace}}(\omega / D)$ the second order expansion of the MAP loss function around ω^{MAP}.

- The Laplace approx. is a cheap but powerful way to turn your MAP model into a Bayesian one through additional approximations:

  1 – Generalized Newton Matrix G: Convex Loss function
  (good approximation in linear regime)

  2 – Fisher Matrix approximation: Kronecker, Diagonal, etc., in order to invert F (or G)

  3 – predictive approximation (linear regime + probit for classification)

  4 – Sub-network approximation

- Better calibrated classifiers and access to the predictive probability distribution, almost for free for a MLE model.

Thanks for your attention !!!