Probabilistic Machine Learning
Lecture 18
Uses for Uncertainty in Deep Learning

Philipp Hennig

3 July 2023

Eberhard Karls Universität Tübingen

Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning

Recap: Deep networks *are* GPs

Linearized networks and Laplace approximations: From deep learning to GPs, in four easy steps

1. Realise that the loss is a **negative log-posterior**

$$\mathcal{L}(\boldsymbol{\theta}) = \Bigg(\frac{1}{N} \sum_{i=1}^{N} \underbrace{\ell(y_i; \overbrace{f(x_i, \boldsymbol{\theta})}^{\text{deep net}})}_{\text{empirical risk}} + \underbrace{r(\boldsymbol{\theta})}_{\text{regularizer}} \Bigg) = -\sum_{i=1}^{N} \log p(\boldsymbol{y} \mid \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta} \mid \boldsymbol{y}) + \text{const.}$$

2. **Train the deep net *as usual*** to find $\boldsymbol{\theta}_* = \arg\max_{\boldsymbol{\theta} \in \mathbb{R}^D} p(\boldsymbol{\theta} \mid \boldsymbol{y})$
3. At $\boldsymbol{\theta}_*$, compute a **Laplace approximation** of the log-posterior, with $\Psi := -\nabla\nabla^\intercal \log p(\boldsymbol{\theta}_* \mid \boldsymbol{y})$

$$\log p(\boldsymbol{\theta} \mid \boldsymbol{y}) + \text{const.} = \mathcal{L}(\boldsymbol{\theta}) \approx \mathcal{L}(\boldsymbol{\theta}_*) + {}^1\!/_2 (\boldsymbol{\theta} - \boldsymbol{\theta}_*)^\intercal \Psi (\boldsymbol{\theta} - \boldsymbol{\theta}_*) = \log \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_*, -\Psi^{-1})$$

4. **Linearize** $f(\boldsymbol{x}, \boldsymbol{\theta})$ around $\boldsymbol{\theta}_*$, with $[J(\boldsymbol{x})]_{ij} = \frac{\partial f_i(\boldsymbol{x}, \boldsymbol{\theta}_*)}{\partial \theta_j}$ as $f(\boldsymbol{x}, \boldsymbol{\theta}) \approx f(\boldsymbol{x}, \boldsymbol{\theta}_*) + J(\boldsymbol{x}, \boldsymbol{\theta}_*)(\boldsymbol{\theta} - \boldsymbol{\theta}_*)$

$$\text{thus } p(f(\bullet) \mid \mathcal{D}) = \int p(f \mid w)\, dp(w) \approx \mathcal{GP}(f(\bullet); f(\bullet, \boldsymbol{\theta}_*), -J(\bullet)\Psi^{-1}J(\circ)) \quad \text{with}$$

$\mathbb{E}(f(\bullet)) = f(\bullet, \boldsymbol{\theta}_*)$ — the **trained net** as the **mean function**

$\operatorname{cov}(f(\bullet), f(\circ)) = -J(\bullet)\Psi^{-1}J(\circ)^\intercal$ — the **Laplace tangent kernel** as the **covariance function**

Computing the *exact* Hessian Ψ is $\mathcal{O}(ND^2)$, and inverting it is $\mathcal{O}(D^3)$.
Ideas for Approximations:

- Sub-sample the dataset ($\mathcal{O}(MD^2)$ with $M \ll N$)
- structural approximations to the Hessian:
  - diagonal approximation: $\mathcal{O}(D)$ (inverse $\mathcal{O}(D)$)
  - last-layer approximation: $\mathcal{O}(D_L^2)$ (inverse $\mathcal{O}(D_L^3)$))
  - Kronecker factorized approximate curvature (KFAC): $\Psi \approx \operatorname{diag}([\Lambda_\ell \otimes \Omega_\ell]_{\ell=1,\dots,L})$ with $\Lambda_\ell \in \mathbb{R}^{\mathrm{in}_\ell \times \mathrm{in}_\ell}, \Omega_\ell \in \mathbb{R}^{\mathrm{out}_{\ell-1} \times \mathrm{out}_{\ell-1}}$ and thus inverse $\mathcal{O}\left(\sum_\ell \mathrm{in}_\ell^3 + \mathrm{out}_{\ell-1}^3\right)$)
  - Generalized Gauss-Newton (homework this week): $\Psi \approx \alpha I + GG^\intercal$ with $G \in \mathbb{R}^{D \times M}$
  - approximate eigenvalue decompositions using the Lanczos algorithm (cf. Lecture 13)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Uncertainty in Deep Learning

- fixes (asymptotic and local) overconfidence
- yields the functionality for continual learning
- many other applications not discussed here

Laplace approximations turn deep networks into GPs, inheriting all functionality of GPs

Please cite this course, as

```
@techreport{Tuebingen_ProbML23,
    title =
    {Probabilistic Machine Learning},
    author = {Hennig, Philipp},
    series = {Lecture Notes
        in Machine Learning},
    year = {2023},
    institution = {Tübingen AI Center}}
```