Latent Derivative Bayesian Last Layer Networks

Summary Notes by Max Guo

July 6, 2022

(?) - denotes a lack of familiarity or understanding of a particular concept at time of reading (?) - denotes a confusion as to why the authors included this

1 Information

- **Year**: 2021
- **Conference**: AISTATS
- **Authors**:

Name	Institute
Joe Watson	Technical University Darmstadt
Jihao Andreas Lin	Technical University Darmstadt
Pascal Klink	Technical University Darmstadt
Joni Pajarinen	Technical University Darmstadt, Aalto University
Jan Peters	Technical University Darmstadt

2 Research Problem

Bayesian Last Layer (BLL) models have overconfident predictions outside of the data distribution.

3 Existing Approaches and Shortcomings

- **BNNs**
  - Intractable inference $\rightarrow$ use approximate inference
  - **Approximate Inference Drawbacks**:
    - Unintuitive priors, expensive training, inaccurate posteriors, large model parameter spaces
  - Inaccurate uncertainty quantification.
- **Gaussian Processes (GPs)**
  - **Drawbacks**:
    - Exact computation does not scale well
    - Some kernels suffer from curse of dimensionality
  - Sparse methods improve scalability (...but we still like parametric models?)
- **BLLs**
  - Neural network learns features, then apply a Bayesian Linear Regression

  - Trained using type-II maximum likelihood.
  - **Drawbacks**:
    * Overparameterization leads to overfitting, limiting predictive uncertainty

4 High Level Contribution

The authors impose a functional prior in the BLL model that involves the model's Jacobian with respect to the inputs to improve the calibration of the epistemic uncertainty expressed by the model.

5 Technical Contributions

5.1 BLL

- *GBLL* = Gaussian process with linear kernel
- *TBLL* = Place inverse gamma prior on σ^2 (noise variance), obtaining Student-t weight posterior and predictive distribution.

5.2 Latent Derivative Priors

- Derivatives of a GP are also a GP (?)
- **Main Idea**: Place functional prior π on the derivatives $\mathbf{z}$ through a functional KL:

$$\min_{\theta} D_{KL}(\pi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x}, \mathcal{D}, \theta)) \tag{1}$$

- Joint training objective:

$$\max_{\theta} \log p(\mathcal{D}|\theta) - D_{KL}(\pi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x}, \mathcal{D}, \theta)) \tag{2}$$

  - Interpreted as maximizing entropy, i.e. choosing diverse features
  - ...or as a latent variable model; objective resembles the ELBO (?)
- Practical Aspects:
  - Functional KL between stochastic processes (e.g. π) is infinite dimensional integral
    * Remedy: Use finite index set $\mathcal{T}$ and evaluate divergence there. Authors use noisy perturbations of the training dataset (OOD)
    * Estimate of the LD fKL, using $\mathcal{T} = \{s_j \sim \mathcal{N}(\cdot|x_j, \gamma I)\}_{j=1}^{n}$:

$$\frac{1}{|\mathcal{T}|} \sum_{s_j \in \mathcal{T}} D_{KL}(\pi(z|s_j)||p(z|s_j, \mathcal{D}, \theta)) \tag{3}$$

  - Choose the latent derivative prior to be a GP with $\mu_\pi(x) = \mathbf{0}$ and $\Sigma_\pi(x) = \mathbf{I}$. (use domain knowledge to set this)
  - Scaling LD prior with aleatoric uncertainty reduces unfitting. (Future work: alternative approaches to specifying prior)

6 Experimentation

- Tasks: Nonlinear regression, Active learning, Bayesian optimization
- **Nonlinear Regression**
  - Benchmarks: standard BLL, nonparametric GP, regularized network (MAP), BNN approaches (MFVI, Monte Carlo Dropout, Ensembles, SWAG)
  - Tasks:
    * "Gap": Cartpole, CO2, Sarcos, WAM
    * "Standard": UCI
  - Results:
    * In the gap tasks, LDBLL outperforms standard BLL in terms of test log likelihood
    * In the standard regression, results were comparable (OOD uncertainty not useful)
    * GP, MC dropout, and Ensembles performed better on both gap and standard regression tasks
      - Authors raise questions about how to design the LD prior.
- **Active Learning**
  - Datasets: Cartpole
  - LDTBLL matches GP (RMSE and Log likelihood), outperforms standard BLL.
- **Bayesian Optimization**
  - Datasets: Sinc in a Haystack (toy, $f(x) = \text{sinc}(6(x-1))$), Hartmann6 (standard BO benchmark)
  - Results:
    * Sinc - LDTBLL outperforms TBLL
    * Hartmann6 - GP is superior and converges faster than LDBLL and BLL. LDBLL converges faster than BLL but both to converge suboptimal values.

7 Further Work

- Specification of the LD prior on a given task or dataset.
- Multivariate prediction tasks (model-based control, classification)

8 My Questions and Thoughts

- This is quite similar to LUNA, the authors modify the training objective in order to increase diversity of the functions via variance in the gradient at a set of points in the data distribution.