Seminar: Statistical Inference in Data Science

Challenges for Statistical Inference in Deep Neural Networks and Stochastic Gradient Descent

Department of Statistics
Ludwig-Maximilians-Universität München

Tobias Brock

Munich, February 6th, 2024

Supervised by Prof. Dr. David Rügamer

Abstract

Statistical inference for deep neural networks and stochastic gradient descent (SGD) faces many challenges. With the wide-spread adoption, understanding the validity of the parameters obtained during training of a complex deep neural network (DNN) and the corresponding predictions, becomes crucial. Due to high-dimensional, non-convex optimization landscapes, assumptions of classical inference cannot be fulfilled. Methods like deep distributional regression try to capture the aleatoric uncertainty of the data, whereas the Jackknife+ is a model agnostic resampling method that estimates the epistemic uncertainty by creating prediction intervals. Capturing parameter uncertainty, on the other hand, is even more challenging. DNNs are typically trained with stochastic optimizers like SGD, which are highly dependent on their initialization and can therefore lead to drastically different results. There exist approaches that try to conduct approximate inference on DNNs and SGD. This paper will discuss three of these approaches and their underlying assumptions in detail. Laplace Redux offers a post hoc transformation of a DNN to a Bayesian neural network (BNN) to enable inference on parameters and predictions that can also be interpreted from a frequentist perspective. For semi-stuctured regression (SSR) models, a framework has been developed that incorporates DNN uncertainty into the theory of generalized linear mixed models as random offset, strictly depending on the quality of the deep uncertainty quantification (DUQ). This approach allows the construction of frequentist confidence intervals for the structured coefficients. Further, for SGD, confidence intervals and classic hypothesis testing for convex objectives and specific learning rate assumptions were derived, using averaged SGD (ASGD). The Laplace or other approximation methods like stochastic low-rank approximate natural-gradient (SLANG) can be used for DUQ, potentially improving the inference framework for SSR models. Moreover, SGD inference can potentially be applied to the convex optimization of the structured coefficients of the SSR models using ASGD. Hence, not requiring a closed form solution or second order methods. All appoaches share a common challenge, namely, the tradeoff between meaningful covariance approximations and computational efficiency. Future research should focus on further improving meaningful covariance approximation methods and extending the theoretical framework of DNNs. Moreover, making SGD inference applicable to DNNs will require generalizing the theory to non-convex objective functions and adaptive learning rates.

Contents

List of Figures

List of Acronyms

ADAM adaptive moment estimation

ASGD averaged SGD

BBP Bayes-by-backprop

BNN Bayesian neural network

DNN deep neural network

DUQ deep uncertainty quantification

ER empirical risk

FNN frequentist neural network

GAMs generalized additive models

GLM generalized linear model

GLMM generalized linear mixed model

KFAC Kronecker-factored approximate curvatur

LA Laplace approximation

MAP maximum a posteriori

MCD Monte Carlo dropout

MCMC Markov chain Monte Carlo

SGD stochastic gradient descent

SLANG stochastic low-rank approximate natural-gradient

SSR semi-stuctured regression

VI variational inference

1 Introduction

Statistical inference in deep neural networks (DNNs) is a challenging task. Non-linearities, induced through non-linear transformations in high-dimensional spaces, make interpretation of effects almost intractable. Although interpretability is not a requirement to conduct valid statistical inference, as seen with Gaussian processes (Liu et al., 2020, Rasmussen and Williams, 2006), the highly non-convex, multimodal optimization landscapes and general black-box nature of DNN problems render classical inference approaches (Lehmann and Romano, 2007, Casella and Berger, 2002) infeasible. Moreover, randomness introduced through stochastic optimizers further exacerbates this problem, as the optimization trajectory is crucially dependent on the initialization and hyperparameter configurations (Tian et al., 2023). Furthermore, in contrast to the generalized linear model (GLM) framework or Gaussian processes, there exists no well-established theory for statistical inference in DNNs (Goan and Fookes, 2020). In DNNs, inference typically requires several strong assumptions and approximations to underlying distributions. Methods like deep distributional regression (Rügamer et al., 2023) attempt to capture the aleatoric uncertainty inherent to the data. The Jackknife+ (Barber et al., 2021), is a model-agnostic resampling technique that creates prediction intervals to capture the epistemic uncertainty of the model. However, these approaches do not provide actual uncertainty estimates for the parameters themselves.

Bayesian neural networks (BNNs) treat parameters as random variables and provide a posterior distribution obtained during the training process. In comparison, a standard frequentist neural network (FNN) uses the optimized weights from the training process as point estimate for the population quantities and does not provide an inherent uncertainty estimate for the predictions. Nonetheless, BNNs are not commonly used in practice due to the difficult implementation and training procedure. Calculating and inverting the covariance matrix of a DNN is computationally infeasible in high dimensional spaces, because the number of parameters can range into the billions. Markov chain Monte Carlo (MCMC) sampling becomes extremely inefficient in such high-dimensional spaces. Methods like stochastic gradient MCMC make training for large amounts of data more efficient, but can lead to poor approximations of the posterior (Goan and Fookes, 2020). Therefore, non-sampling-based approximations of the posterior distribution, such as the Laplace approximation (LA) and variational inference (VI), which require the estimation of the covariance matrix, are essential. Methods like the LA can be applied post hoc after training to transform a FNN into a highly functioning BNN (Daxberger et al., 2021). Further, the uncertainty of a DNN may be captured indirectly using semi-structured regression (SSR) models, where the DNN is incorporated as random offset (Dorigatti et al., 2023). Last but not least, SGD inference provides frequentist confidence intervals and enables hypothesis testing for averaged SGD (ASGD) estimates performed on convex objective functions (Singh et al., 2023). The main part of this paper will discuss these three approaches and how they can potentially be combined in detail. Two key challenges are identified: (1) Inference on DNNs requires meaningful approximations to the Hessian matrix. (2) The general lack of theory requires indirect inference techniques on the DNN structure and strong distributional assumptions on the Hessian matrix.

2 Current Research

Because there is no theoretical framework to conduct statistical inference directly on FNNs, methods like Laplace Redux (Daxberger et al., 2021) aim to approximate the posterior distribution of the parameters post hoc, after regular FNN training, to transform the network into a BNN, using a Laplace approximation. Another way to conduct inference indirectly on a DNN, as described by Dorigatti et al. (2023), involves incorporating the uncertainty of a DNN into the additive predictors of their SSR models, using the theoretical framework of the generalized linear mixed model (GLMM) and frequentist theory. A framework by Singh et al. (2023), applicable to convex optimization objectives, allows for generating asymptotically normal and consistent estimators of the covariance matrix for an equal batch-size strategy in SGD. Hence, enabling the application of frequentist theory. This section will provide a detailed overview of these approaches, discuss their advantages and shortcomings, and explore how they can potentially be combined.

2.1 Laplace Redux

The Laplace approximation is a simple family of approximation algorithms for the posterior distribution. It uses a Gaussian distribution centered at a local maximum to approximate the posterior, with the local curvature as covariance matrix. Daxberger et al. (2021) propose the LA as cost-efficient and competitive approximation method for statistical inference in BNNs. The local maximum obtained by standard FNN training, corresponding to the maximum a posteriori (MAP) estimate from a Bayesian perspective, is directly available. Subsequently, the local curvature can be estimated by efficient approximations, due to advances in second-order optimization and easy-to-use software libraries of the Hessian, evaluated at the MAP estimate. This enables the post hoc transformation of a powerful FNN to a BNN that provides uncertainty quantification of its parameters and predictions. Moreover, the model evidence can be selected to enable model selection via hyperparameter tuning.

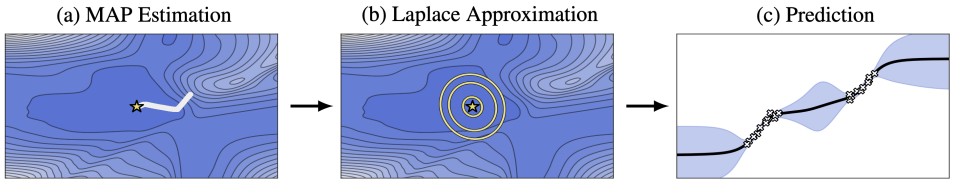

Figure 1: Daxberger et al. (2021). The MAP estimate is found via basic training, then the posterior landscape is approximated with a local Gaussian centered at the MAP estimate. Afterward, the LA can be used to provide predictive uncertainty estimates but also for parameter inference.

Consider the empirical risk minimization set up of a supervised deep learning problem. E.g., given an i.i.d. classification dataset $\mathcal{D} := \left\{\left(x_n \in \mathbb{R}^M, y_n \in \mathbb{R}^C\right)\right\}_{n=1}^N$, weights $\theta \in \mathbb{R}^D$ of an L-layer DNN, a function $f_\theta : \mathbb{R}^M \to \mathbb{R}^C$ is trained to minimize the (usually regularized) empirical risk (ER). The ER is normally decomposed into a sum of individual

loss terms $\ell(x_n, y_n; \theta)$ and a regularization penalty $r(\theta)$. Minimizing the regularized ER is equivalent to finding the MAP estimate of θ.

$$\theta_{\text{MAP}} = \arg\min_{\theta \in \mathbb{R}^D} \mathcal{L}(\mathcal{D}; \theta) = \arg\min_{\theta \in \mathbb{R}^D} \left(r(\theta) + \sum\nolimits_{n=1}^{N} \ell(x_n, y_n; \theta) \right). \tag{1}$$

Where the individual loss terms can be identified as the i.i.d. log-likelihood terms and the regularization penalty as the log-prior, yielding $\ell(x_n, y_n; \theta) = -\log p(y_n | f_\theta(x_n))$ and $r(\theta) = -\log p(\theta)$. One of the most common forms of regularization in DNNs is a weight decay regularizer $r(\theta) = \frac{1}{2}\gamma^{-2}||\theta||^2$, which corresponds to a centered Gaussian prior $p(\theta) = \mathcal{N}(\theta; 0, \gamma^2 I)$ from a Bayesian perspective and a categorical likelihood when the cross-entropy loss is used. Taking the negative exponential of both terms results in an unnormalized posterior, e.g., with the cross-entropy loss. The normalized posterior distribution is then given by

$$p(\theta|\mathcal{D}) = \frac{1}{Z} p(\mathcal{D}|\theta) p(\theta), \;\; Z := \int p(\mathcal{D}|\theta) p(\theta) d\theta, \tag{2}$$

with intractable normalizing constant Z, also referred to as marginal likelihood or evidence. The LA uses a second-order expansion of $\mathcal{L}$ around the MAP estimate to build a Gaussian approximation to the posterior $p(\theta|\mathcal{D})$. Consider the expansion

$$\mathcal{L}(\mathcal{D}; \theta) \approx \mathcal{L}(\mathcal{D}; \theta_{\text{MAP}}) + \frac{1}{2}(\theta - \theta_{\text{MAP}})^T \left(\nabla^2_\theta \mathcal{L}(\mathcal{D}; \theta)|_{\theta_{\text{MAP}}} \right) (\theta - \theta_{\text{MAP}}) \tag{3}$$

with the first order term vanishing at the point θ_{MAP}. The LA is therefore given by

$$p(\theta|\mathcal{D}) \approx \mathcal{N}(\theta; \theta_{\text{MAP}}, \Sigma), \;\; \Sigma := \left(\nabla^2_\theta \mathcal{L}(\mathcal{D}; \theta)|_{\theta_{\text{MAP}}} \right)^{-1}. \tag{4}$$

The normalizing constant Z can be used for model selection and is approximated by

$$Z \approx \exp(-\mathcal{L}(\mathcal{D}; \theta_{\text{MAP}}))(2\pi)^{D/2} (\det \Sigma)^{1/2}. \tag{5}$$

2.1.1 Hessian Approximations

Since θ_{MAP} is obtained by standard ER minimization, the only additional step for the LA is the computation of the inverse Hessian matrix (4) evaluated at θ_{MAP}, which can be conducted post hoc after training. The paper assumes a normal prior $p(\theta) = \mathcal{N}(\theta; 0, \gamma^2 I)$, due to popularity of weight decay regularization in DNNs. This results in the Hessian depending on the log-prior and the empirical risk

$$\nabla^2_\theta \mathcal{L}(\mathcal{D}; \theta)|\theta_{\text{MAP}} = -\gamma^{-2} I - \sum\nolimits_{n=1}^{N} \nabla^2_\theta \log p(y_n | f_\theta(x_n))|_{\theta_{\text{MAP}}}. \tag{6}$$

It is straightforward to see that the Hessian scales quadratically in the number of parameters because of $\nabla^2_\theta \log p(y_n | f_\theta(x_n))|_{\theta_{\text{MAP}}}$, possibly ranging in the billions for DNNs, rendering a naive implementation of the Hessian infeasible. Therefore, efficient approximations of the Hessian are required to perform inference over all parameters. LA can benefit from positive semi-definite approximations to the potentially indefinite Hessian matrix, i.e., with the Fisher information matrix or simply Fisher

$$F := \sum\nolimits_{n=1}^{N} \mathbb{E}_{\hat{y} \sim p(y|f_\theta(x_n))} \left[(\nabla_\theta \log p(y|f_\theta(x_n))|_{\theta_{\text{MAP}}}) (\nabla_\theta \log p(y|f_\theta(x_n))|_{\theta_{\text{MAP}}})^T \right], \tag{7}$$

or the generalized Gauss-Newton matrix (GGN)

$$G := \sum_{n=1}^{N} (\nabla_\theta f_\theta(x_n))|_{\theta_{\text{MAP}}}) \left(\nabla_f^2 \log p(y_n|f)|_{f=f_{\theta_{\text{MAP}}}(x_n)} \right) (\nabla_\theta f_\theta(x_n))|_{\theta_{\text{MAP}}})^T, \quad (8)$$

which are equivalent for common log-likelihoods. Since both F and G scale quadratically with the number of parameters, further factorization assumptions are required. A simple, yet naive assumption is a diagonal factorization, as it ignores off-diagonal elements (assuming independence). A more expressive choice might be a block-diagonal factorization like Kronecker-factored approximate curvatur (KFAC) or low-rank approximations of the Hessian or Fisher, as for example proposed by Mishkin et al. (2018). They provide a fast structured covariance approximation for BNNs with natural gradients. The method can be competitive with VI, which is commonly used to approximate the posterior distribution, typically with a Gaussian, when training a BNN. The parameters are learned by backpropagating natural gradients, necessitating the storage and inversion of the $D \times D$ covariance matrix for each iteration. As previously discussed, this calculation is infeasible for DNNs due to computational complexity. Therefore, they estimate a low-rank plus diagonal approximation of the covariance matrix Σ that is cost-efficient, but preserves some off-diagonal structure to model dependencies between the weights. The SLANG procedure generally approximates the posterior with a Gaussian distribution and trains a BNN from scratch, whereas LA can also be applied post hoc. Different hessian approximations are visualized in figure 2.

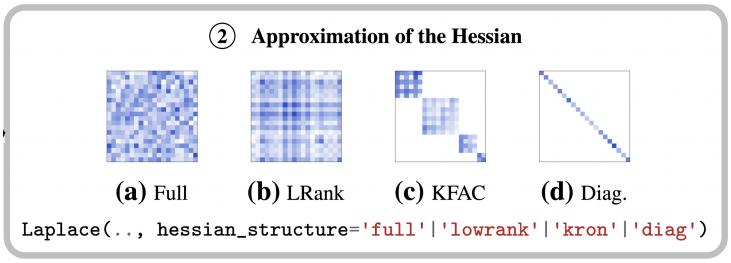

Figure 2: Daxberger et al. (2021). Covariance structure of different Hessian approximations.

2.1.2 Subnetwork Inference

The authors further suggest application of the LA, without using a Hessian approximation to efficiently scale their approach to large DNN. Instead of storing the intractable $D \times D$ covariance matrix, one may only perform inference over a small subset of model parameters. This approach is based on recent evidence that suggests DNNs can be pruned extensively without a substantial loss of test accuracy (Frankle and Carbin, 2019). Moreover, in the neighborhood of a local minimum, the predictions remain unchanged for a large set of directions (Maddox et al., 2020). For the subnetwork inference, the posterior is approximated by

$$p(\theta|\mathcal{D}) = p(\theta_S|\mathcal{D}) \prod_r \delta(\theta_r - \hat{\theta}_r) = q_S(\theta), \quad (9)$$

with $\delta(\theta_r - \hat{\theta}_r)$ denoting the Dirac delta function centered at the predicted parameter vector $\hat{\theta}_r$ of the remaining $D - S = r$ weights that are held constant at their MAP estimated values. The term $p(\theta_S|\mathcal{D})$ is a Laplace posterior over the subnetwork $\theta_S \in \mathbb{R}^S$, with the subnetwork size S being a hyperparameter, and $S \ll D$ such that it becomes tractable to store the entire $S \times S$ covariance matrix Σ_S. This enables the exploitation of the full subnetwork covariance structure, as no further independence or factorization assumptions are required. If Σ_S is still intractable, one may simply perform Hessian approximations as previously discussed.

A special case of subnetwork inference is last layer Laplace, where $\theta_S = W^{(L)}$ is the weight matrix of the last network layer, while holding the remaining weights constant at their MAP estimated values. The Laplace approximated posterior over the last layer is given by

$$p\left(W^{(L)}|\mathcal{D}\right) \approx \mathcal{N}\left(W^{(L)}|W^{(L)}_{\text{MAP}}, \Sigma^{(L)}\right), \tag{10}$$

which results in a small covariance matrix relative to the entire network, enabling efficient approximation. Figure 3 shows the different inference approaches. As stated previously,

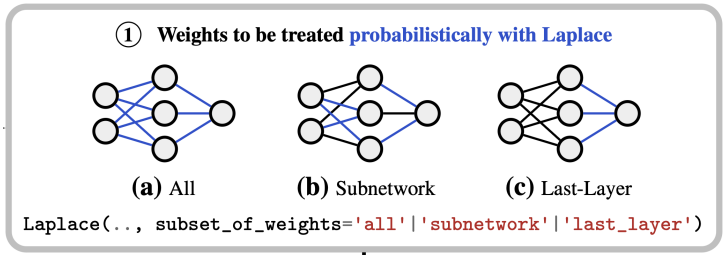

Figure 3: Daxberger et al. (2021). Inference for different sets of network weights.

observe that θ_{MAP} corresponds to the regularized risk minimizer θ_{reg}. Given the LA, one may simply assume that

$$p(\theta|\mathcal{D}) \approx \mathcal{N}(\theta; \theta_{\text{MAP}}, \Sigma) = \mathcal{N}(\theta; \theta_{\text{reg}}, \Sigma), \Sigma := \left(\nabla^2_\theta \mathcal{L}(\mathcal{D}; \theta)|_{\theta_{\text{reg}}}\right)^{-1}, \tag{11}$$

which can be used to construct an approximate frequentist confidence interval for θ_{reg}, given an appropriate approximation of the inverse Hessian. Notice that θ_{reg} is a biased estimate of θ due to the regularization penalty and the inherent complexity of the DNN. Thus, the LA can be interpreted from a frequentist perspective by simply changing the point of view on the optimization problem. Therefore, the quality of the inference is mostly dependent on how well the covariance matrix is estimated and on the suitability of the multivariate normal approximation for the posterior distribution.

2.2 Semi-Structured Regression

Semi-structured regression models allow to jointly learn the effect of structured (tabular) and unstructured (non-tabular, e.g., images) data by incorporating an additive predictor and a DNN into a single model. Dorigatti et al. (2023) have developed a method that enables classic statistical inference by using the framework of GLMMs on the structured

coefficients of the additive predictor, while simultaneously incorporating the uncertainty of the DNN as random offset. This approach leads to a decrease in the Type-I error rate, as it accounts for the higher variance induced by the DNN. To conduct inference on the structured coefficients, a two-stage fitting procedure is considered. (1) The DNN and additive predictor are jointly learned on a training set, with a second, independent held-out dataset for early stopping. (2) Inference on the structured coefficients is then performed on the validation set, incorporating the DNN predictions and their associated uncertainty.

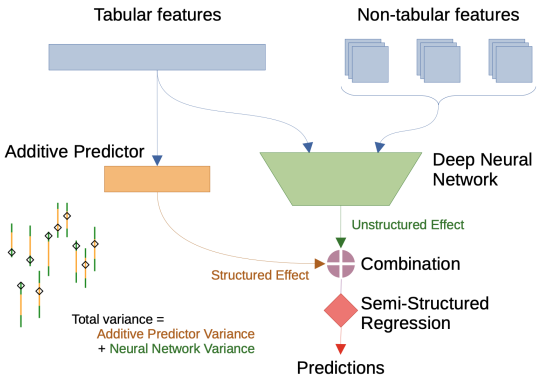

Figure 4: Dorigatti et al. (2023). Structure of a SSR model.

Denote the design matrix of the tabular features $\mathbf{x}_i \in \mathbb{R}^d$ by $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{X} = \left(\mathbf{x}_1^T, ..., \mathbf{x}_n^T\right)^T$ for the n validation samples. Let $\boldsymbol{\beta} \in \mathbb{R}^d$ and $\boldsymbol{f} \in \mathbb{R}^n$ be the true parameter of the additive predictor and the true additive effect of the non-tabular features, respectively. Then, the predictions for the response $\mathbf{y} = (y_1, ..., y_n)^T \in \mathbb{R}^n$ are calculated by the linear predictor $\eta := \mathbf{X}\boldsymbol{\beta} + \boldsymbol{f}$, where $\boldsymbol{f}$ is separately estimated by the DNN predictions $\mathbf{z} \in \mathbb{R}^n$, $\mathbf{z} \sim \mathcal{N}(\boldsymbol{f}, \boldsymbol{\Gamma})$, with known full-rank covariance matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{n \times n}$.

2.2.1 Linear Models

Consider the data-generating process $\mathbf{y} = \mathbf{X}\beta + \boldsymbol{f} + \boldsymbol{\epsilon}$, with $\mathbf{z} \sim \mathcal{N}(\boldsymbol{f}, \boldsymbol{\Gamma})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, where $\mathbf{z}$ is again the DNN prediction and $\boldsymbol{\beta}$ the coefficient vector of the additive predictor. Then, an unbiased estimator of $\boldsymbol{\beta}$ is straightforwardly calculated by

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^\mathbf{T}\mathbf{X}\right)^{-1} \mathbf{X}^T(\mathbf{y} - \mathbf{z}), \tag{12}$$

with variance

$$\mathbb{V}[\hat{\boldsymbol{\beta}}] = \left(\mathbf{X}^\mathbf{T}\mathbf{X}\right)^{-1} \mathbf{X}^T \left(\sigma^2 \boldsymbol{I} + \boldsymbol{\Gamma}\right) \mathbf{X} \left(\mathbf{X}^\mathbf{T}\mathbf{X}\right)^{-1}. \tag{13}$$

An unbiased estimator of σ^2, given the residuals $\boldsymbol{r} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{z}$ is

$$\hat{\sigma}^2 = \left(\boldsymbol{r}^T\boldsymbol{r} - tr(\boldsymbol{\Gamma})\right) / (n - d). \tag{14}$$

Notice that the variance of $\hat{\boldsymbol{\beta}}$ corresponds to the ordinary least squares estimator with heteroscedastic errors. Moreover, for $\sigma^2 \gg tr(\boldsymbol{\Gamma})/n$, the DNN uncertainty has a minor

effect as most of the variance stems from $\boldsymbol{\epsilon}$. For $\boldsymbol{\Gamma} \to diag\,(\gamma^2/n \cdot 1)$, meaning constant uncertainty γ^2 from the DNN (when, e.g., the additive predictor is refitted on a seperate held-out dataset), the variance of the estimated predictor reduces to $\mathbb{V}[\hat{\boldsymbol{\beta}}] = (\sigma^2 + \gamma^2)\left(\mathbf{X^T X}\right)^{-1}$, where both scalar parameters can be estimated from $\boldsymbol{r}$. This results in confidence intervals of nominal coverage for constant DNN predictions.

2.2.2 Penalized SSR Models

The theory discussed beforehand is easily extendable to penalized least squares, specifically to Ridge regression and additive models. For additive models, let the design matrix $\mathbf{X}$ include suitable basis expansions for all smooth terms and let $\mathbf{S}_{\boldsymbol{\lambda}}$ be the penalty matrix, with $\boldsymbol{\lambda}$ controlling the regularization strength. For Ridge regression, this yields $\boldsymbol{\lambda} = \lambda^2\mathbf{1}$ and hence $\mathbf{S}_{\boldsymbol{\lambda}} = \lambda^2\boldsymbol{I}$, whereas for additive models, one obtains $\mathbf{S}_{\boldsymbol{\lambda}} = \sum_i \lambda_i \mathbf{S}_i$ for each $\lambda_i, \mathbf{S}_i$ controlling the smoothness of the i-th smooth. The inference for additive models is then performed by assuming the same data-generating process as before and defining

$$\hat{\boldsymbol{\beta}} = \mathbf{P}(\mathbf{y} - \mathbf{z}), \tag{15}$$

with $\mathbf{P} := \left(\mathbf{X^T X} + \mathbf{S}_{\boldsymbol{\lambda}}\right)^{-1}\mathbf{X^T}$ and variance

$$\mathbb{V}[\hat{\boldsymbol{\beta}}] = \mathbf{P}\left(\sigma^2\boldsymbol{I} + \boldsymbol{\Gamma}\right)\mathbf{P}^T. \tag{16}$$

σ^2 can be estimated by letting $\boldsymbol{r} = \mathbf{y} - \mathbf{X}\beta - \mathbf{z}$

$$\hat{\sigma}^2 = \left(\boldsymbol{r}^T\boldsymbol{r} - tr(\boldsymbol{\Gamma})\right)/(n - d). \tag{17}$$

Optimal penalties λ can be obtained by cross-validation. Notice, that the estimate for $\hat{\boldsymbol{\beta}}$ is no longer unbiased due to the regularization penalty. Therefore, not allowing an easy calculation of prediction intervals. However, constructing confidence intervals on the biased coefficients is not forestalled by this approach. But, with the previously described two-stage fitting procedure employed by the authors, bias in the form of artificial shrinking (towards 0) can be corrected for, if $\hat{\boldsymbol{\beta}}$ is re-estimated after training the SSR model on a separate held-out dataset, such as a validation set employed for early stopping. This enables obtaining unbiased coefficients even when penalization is used, thus facilitating the calculation of meaningful confidence intervals.

2.2.3 Generalized SSR Models

Further generalizations are possible for SSR models with an exponential response distribution. Then, the conditional mean of the response can be written as a function of the linear predictor

$$\mathbb{E}[\mathbf{y}|\mathbf{X}] = g^{-1}(\mathbf{X}\beta + \boldsymbol{f}),\ \mathbf{z} \sim \mathcal{N}(\boldsymbol{f}, \boldsymbol{\Gamma}). \tag{18}$$

Now, the DNN's predictions $\hat{\mathbf{z}}$ of $\mathbf{z}$ are fixed and the remaining variation is modeled by a random effect $\boldsymbol{b} \sim \mathcal{N}(0, \boldsymbol{\Gamma})$. The inference in a semi-structured GLM is performed by solving the equivalent formulation of a GLMM

$$\mathbb{E}[\mathbf{y}|\boldsymbol{\beta}, \mathbf{z}] = g^{-1}(\mathbf{X}\beta + \hat{\mathbf{z}} + \boldsymbol{b}), \tag{19}$$

for a response distribution $\mathcal{D}$, some fixed offset $\hat{\mathbf{z}}$ and random effects $\boldsymbol{b}$. $\mathbf{W}$ denotes the GLM weights and ϕ the scale parameter for $\mathcal{D}$. Therefore, $\hat{\boldsymbol{\beta}}$ and $\hat{\mathbf{z}}$ can be found by penalized least squares, which produces the asymptotic relationship

$$\begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{b}} \end{pmatrix} \overset{a}{\sim} \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\beta} \\ \mathbf{0} \end{pmatrix}, \phi \begin{pmatrix} \mathbf{X}^T\mathbf{W}\mathbf{X} & \mathbf{X}^T\mathbf{W} \\ \mathbf{W}\mathbf{X} & \mathbf{W} + \phi\boldsymbol{\Gamma}^{-1} \end{pmatrix}^{-1} \right). \tag{20}$$

These results are extendable to include non-linear structured effects into η, allowing for flexible but interpretable univariate or low-dimensional multivariate smooth effects. Treating smooth terms as random effects offers more flexibility, as it enables the estimation with a linear mixed model solver. Therefore, allowing the extension of the framework to semi-structured generalized additive models (GAMs) and mixed GAMs. Experimental results show that DNN coverage becomes progressively worse when not accounting for the increasing uncertainty of the DNN predictions. On the other hand, nominal coverage can be achieved with the GLMM approach that includes the DNN uncertainty. However, increased DNN influence on the structured coefficients also decreases the power of the test, due to a natural increase in variability. Notice that all three optimization problems are in most cases convex. Proofs for section 2.2.1 and 2.2.2 can be found in the appendix, where the latter is dependent on the penalty matrix $\mathbf{S}_{\boldsymbol{\lambda}}$. For section 2.2.3, the authors mention that the optimization is convex in $\boldsymbol{\beta}$ and $\boldsymbol{b}$. Convexity of the objective is advantageous, as it could allow the application of SGD inference, which is discussed in the next section.

Figure 5 displays nine SSR models, where the additive predictor is fitted to a thin plate regression spline on a simulated dataset (red line). One can observe that the predictors are moderately overfitted after initial training with SGD, due to the wiggliness of the estimates (dashed lines). Fitting additive models post hoc by using the DNN predictions, still results in overfitting (dotted lines). Ensembling the additive models provides a better average fit than simply ensembling the SSR models and improved confidence intervals. However, these are still overly narrow in some regions. Incorporating the DNN uncertainty into the structured coefficients provides the best fit (Theorem 3 of the authors). DNN uncertainty is low for dense regions of the data distribution and high at the extremes.

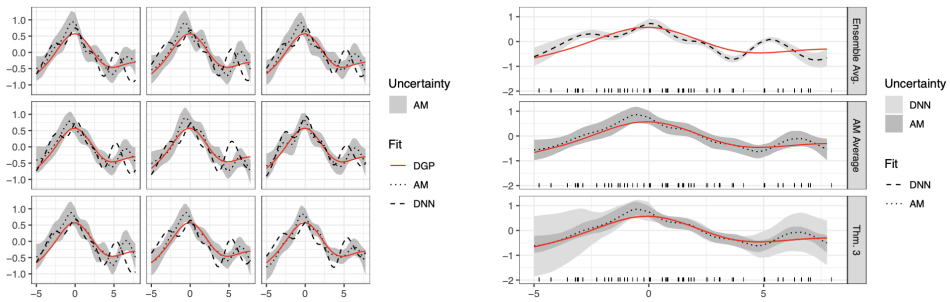

Figure 5: Dorigatti et al. (2023). Confidence regions of different fitting approaches.

Notice that the quality of the uncertainty estimation is strictly dependent on the employed deep uncertainty quantification (DUQ). Inadequate DUQ cannot be offset by the

method proposed by the authors. For their comparison, the coverage of a regular SSR trained with Bayes-by-backprop (BBP), an ensemble of SSR models, two DNN trained with ensembling, and Monte Carlo dropout (MCD) respectively, as two oracle predictors also trained with ensembling and MCD were used. BBP, MCD and ensembles naturally provide uncertainty estimates of $\mathbf{\Gamma}$. To obtain a baseline for DUQ, $\mathbf{\Gamma}$ was discarded from the regular DNNs trained with MCD and ensembling. Then, GLMMs were fitted using MCD and ensembling to compare them with the uncertainty unaware DNNs and the regular SSR models.

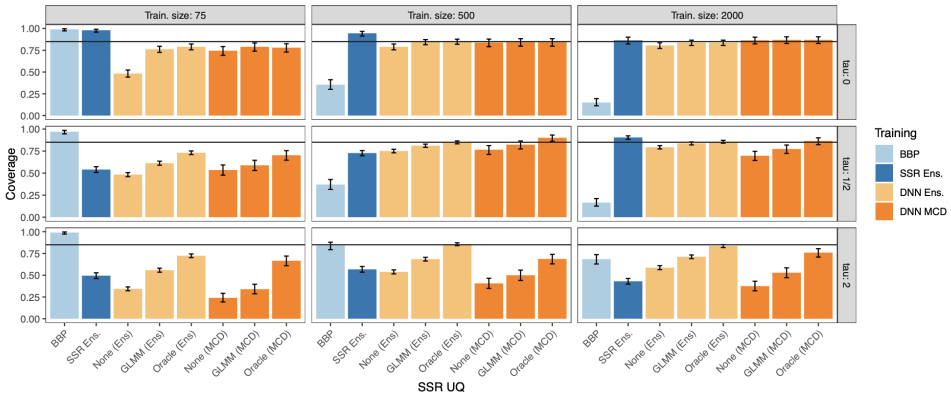

Figure 6: Dorigatti et al. (2023). Coverage of fitting approaches.

Figure 6 shows that for an increased unstructured effect displayed by larger values of τ, DUQ becomes more important. Especially for $\tau = 2$, using the uncertainty of the DNN in the GLMMs provides higher coverage. One may further observe that ensembling outperforms MCD. However, ensembling large DNN models may not always be computationally feasible. Moreover, the SSR trained with BBP performs quite well. BBP is a VI approach that uses mean field approximation to obtain the parameters in the SSR model. As noted by Mishkin et al. (2018), mean field approximations may perform worse in the presence of strong posterior correlations and lead to variance shrinkage. Using SLANG may further improve the DUQ of both the standard SSR and the GLMMs. Furthermore, the LA is another viable method for obtaining DUQ and could be employed for more versatile comparisons with a powerful FNN, potentially decreasing the predictive uncertainty of the DNN itself in comparison to a BNN.

2.3 SGD Inference

SGD is a stochastic optimization technique that can handle large amounts of data, as it does not require storing full-size gradients, which is infeasible in the optimization of DNNs (Tian et al., 2023). Conducting inference for SGD is not straightforward. Although SGD provides unbiased estimates of the real gradient under mild conditions, SGD is highly stochastic in its nature. Different initializations of the algorithm result in different optimization trajectories. Further, the choice of the learning rate is crucial for successful convergence and can therefore lead to drastically varying results. SGD by its nature is Markovian, since the current update only depends on the previous state, where for each

update a random observation or subset of the data is chosen to update the current gradient (Singh et al., 2023, Tian et al., 2023). This renders statistical inference, especially on highly non-convex optimization surfaces, a difficult task.

Singh et al. (2023) show that the normality of an averaged SGD (ASGD) estimator allows the construction of a batch-means estimator for the asymptotic covariance matrix. For convex objective functions, an equal batch-size strategy provides consistency of the covariance matrix, thereby allowing bias-correction and marginal friendly simultaneous confidence intervals. Additionally, the asymptotic normality and consistency allow for traditional hypothesis testing.

Formally, let Π be a probability distribution on $\mathbb{R}^r$, with data ζ arising from said distribution, $\zeta \sim \Pi$. The empirical loss for the estimation of a parameter vector $\theta \in \mathbb{R}^d$ can be denoted by a function $f : \mathbb{R}^d \times \mathbb{R}^r \to \mathbb{R}$. The expected loss is written as $F(\theta) = \mathbb{E}_{\zeta \sim \Pi}[f(\theta, \zeta)]$. The optimizer is then given by

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^d} F(\theta). \tag{21}$$

For $\zeta_i \overset{iid}{\sim} \Pi$, $i = 1, ..., n$ one wants to obtain θ^*. Usually, $F(\theta)$ is approximated by the empirical loss $n^{-1} \sum_{i=1}^{n} f(\theta, \zeta_i)$. For large datasets, the gradient is then estimated with an unbiased estimate. Let $\nabla f(\theta, \zeta)$ be the gradient vector of f with respect to θ. Denote by $\eta_i > 0$ the learning rate, and θ_0 as starting point. The i^{th} iteration of SGD is thus

$$\theta_i = \theta_{i-1} - \eta_i \nabla f(\theta_{i-1}, \zeta_i), \ i = 1, 2, \ldots . \tag{22}$$

For η_i appropriately decreasing and θ^* being the ASGD estimator $\hat{\theta}_n := n^{-1} \sum_{i=1}^{n} \theta_i$, nice statistical properties can be derived. Let $A := \nabla^2 F(\theta^*)$ be the Hessian of the objective function $F(\theta)$ evaluated at θ^*, and define the expected value of the outer product of the gradients as $S := \mathbb{E}_{\Pi}\left([\nabla f(\theta^*, \zeta)] [\nabla f(\theta^*, \zeta)]^T \right)$. For F strictly convex with Lipschitz gradient and $\eta_i = \eta i^{-\alpha}, \alpha \in (0.5, 1)$, $\hat{\theta}_n$ is a consistent estimator of θ^*, and for some additional conditions it holds that

$$\sqrt{n}\left(\hat{\theta}_n - \theta^*\right) \xrightarrow{d} N(0, \Sigma) \text{ as } n \to \infty, \text{ where } \Sigma = A^{-1} S A^{-1}. \tag{23}$$

Valid inference depends crucially on the estimation of Σ. The authors employ an equal batch-sizes strategy for practical utility and theoretical guarantees. This strategy yields a consistent estimator and mean square-error bounds under mild conditions. Further, it allows bias-reduction of Σ, which is typically under-biased due to the Markovian structure of the SGD estimates.

2.3.1 Batch-Means Estimator

For iteration size n, the SGD iterations are divided into K batches with sizes $b_{n,1}, ..., b_{n,K}$ after some specified warm-up period. Let $\tau_k = \sum_{j=1}^{k} b_{n,j}$ for $k = 1, ..., K$ be the ending index of the kth batch

$$\underbrace{\{\theta_1, ..., \theta_{\tau_1}\}}_{1^{\text{th}} \ batch}, ..., \underbrace{\{\theta_{\tau_{K-1}+1}, ..., \theta_{\tau_K}\}}_{K^{\text{th}} \ batch}. \tag{24}$$

Denote the mean vector of batch k by $\overline{\theta}_k = b_{n,k}^{-1} \sum_{i=\tau_{k-1}+1}^{\tau_k} \theta_i$. This yields a general batch means estimator

$$\hat{\Sigma}_{\text{gen}} = \frac{1}{K} \sum_{k=1}^{K} b_{n,k} (\overline{\theta}_k - \hat{\theta}_n)(\overline{\theta}_k - \hat{\theta}_n)^T. \tag{25}$$

For an equal batch-size strategy, let $b_{n,k} = b_n$ for all k and $a_n := K = [n/b_n]$ be the number of batches. Consequently, resulting in the estimator

$$\hat{\Sigma}_{b_n} = a_n^{-1} \sum_{k=1}^{a_n} b_n (\overline{\theta}_k - \hat{\theta}_n)(\overline{\theta}_k - \hat{\theta}_n)^T = \frac{b_n}{a_n} \sum_{k=1}^{a_n} \overline{\theta}_k \overline{\theta}_k^T - b_n \hat{\theta}_n \hat{\theta}_n^T, \tag{26}$$

by simply noting that $\sum_{k=1}^{a_n} \overline{\theta}_k = a_n \hat{\theta}_n$. The authors employ several assumptions, e.g., the objective function $F(\theta)$ being continuously differentiable and strongly convex (not applicable to DNNs) and fixing the learning rate to $\eta_i = \eta i^{-\alpha}, \alpha \in (0.5, 1), i = 1, 2, 3, \ldots$. As a consequence, it holds that

$$\sum_{i=\tau_{k-1}+1}^{\tau_k} \eta_i = \sum_{i=\tau_{k-1}+1}^{\tau_k} \eta i^{-\alpha} > \eta b_n \tau_k^{-\alpha} > \eta b_n n^{-\alpha} =: N. \tag{27}$$

Choosing the batch-size b_n, such that $b_n n^{-\alpha} \to \infty$ and $b_n n^{-1}$ as $n \to \infty$, with $b_n = cn^\beta$ for some $\beta \in (\alpha, 1)$ and $c > 0$, yields that $N \to \infty$ as $n \to \infty$. This condition provides that the batch-size must be larger than the persistent correlation of SGD iterates, ensuring fast decay of correlations between the batches.

For some constant C_d depending on d, the assumptions of the authors, and sufficiently large n, one obtains

$$\mathbb{E}||\hat{\Sigma}_{b_n} - \Sigma|| \lesssim C_d^{3/2} n^{-\alpha/4} + C_d^{3/2} a_n^{-1/2} + C_d b_n^{\alpha-1} + C_d b_n^{-1/2} n^{\alpha/2} + C_d a_n^{-1} + C_d^4 n^{-2\alpha} b_n, \tag{28}$$

with the right-hand side going to 0 for the assumed batch-size and learning rate yielding the consistency of $\hat{\Sigma}_{b_n}$. This bound allows for reasonable choice of b_n. Having $b_n = cn^\beta$ gives

$$\mathbb{E}||\hat{\Sigma}_{b_n} - \Sigma|| \lesssim n^{-\alpha/4} + n^{(\beta-1)/2} + n^{-\beta(1-\alpha)} + n^{(\alpha-\beta)/2} + n^{\beta-1} + n^{\beta-2\alpha}. \tag{29}$$

Noting that $(\beta - 1)/2 > \beta - 2\alpha$ and $-\beta(1-\alpha) < (\alpha - \beta)/2$, only the dominating terms can be considered

$$\mathbb{E}||\hat{\Sigma}_{b_n} - \Sigma|| \lesssim n^{-\alpha/4} + n^{(\beta-1)/2} + n^{(\alpha-\beta)/2}. \tag{30}$$

Differentiating this approximation with respect to β gives the optimal choice $\beta^* = (1 + \alpha)/2$ for sufficiently large n. The consistency of $\hat{\Sigma}_{b_n}$ enables the construction of Wald-like confidence regions.

2.3.2 Bias-Correction

The derived bound naturally includes both bias and variance of the estimator. Arguably, the bias of a variance estimator is more important than its variance, especially when the bias is negative, as it leads to inadequate tests. With the lugsail technique for batch-means estimators, the bias can be drastically reduced by partially increasing lag-windows

in spectral variance estimators (Vats and Flegal, 2022). The lugsail estimator is a linear combination of variance estimators, leading to a flexible and consistent class of bias-corrected variance estimators in steady-state simulation and MCMC. The estimator is given by

$$\hat{\Sigma}_{L,b_n} = 2\hat{\Sigma}_{2b_n} - \hat{\Sigma}_{b_n}, \tag{31}$$

which can be easily obtained by combining adjacent batch-means estimators for an equal batch-size strategy. Notice that the bias reduction depends on the correlation of the process, expressed by α. However, the correction is still significant for finite-sample performance, as can be observed in figure 7.

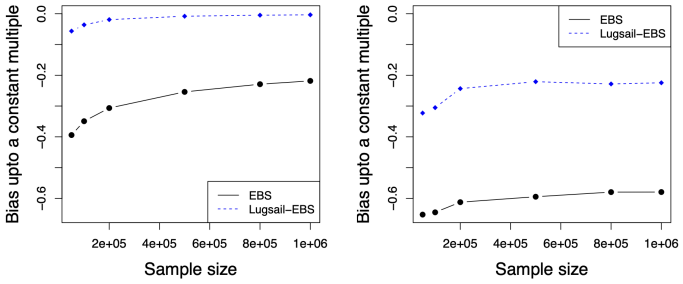

Figure 7: Singh et al. (2023). Bias correction for $\alpha = 0.51$ and $\alpha = 0.75$ respectively in a simple mean estimation model $y_i = \theta^* + \epsilon_i$ with loss function $f(\theta, \zeta) = (y - \theta)^2/2$.

2.3.3 Parameter Inference for SGD

Naive use of the diagonal elements of Σ to construct uncorrected marginal confidence intervals for the d parameters leads to the multiple-testing problem. Corrections like Bonferroni can be notoriously crude, while complex dependencies within Σ are additionally ignored. Moreover, a d-dimensional confidence ellipsoid is not feasible for marginal-friendly inference. Therefore, the authors provide marginal-friendly simultaneous confidence intervals. First, consider a $100(1-p)\%$ confidence ellipsoid for joint inference on θ^*

$$E_p = \left\{\theta \in \mathbb{R}^d : (\hat{\theta}_n - \theta)^T \hat{\Sigma}_n^{-1} (\hat{\theta}_n - \theta) \leq \chi^2_{d,1-p}\right\}, \tag{32}$$

where $\chi^2_{d,1-p}$ is the $(1-p)$th quantile of a chi-squared distribution with d degrees of freedom. Because marginal interpretations are not necessarily feasible, it is beneficial to consider marginal confidence intervals. Denote $\hat{\Sigma}_n$ with $\hat{\Sigma}_n = (\hat{\sigma}_{ij})_{i,j=1,\ldots,p}$ as a consistent estimator of Σ and let $\hat{\theta}_n = (\hat{\theta}_{n1}, ..., \hat{\theta}_{nd}), \theta^* = (\theta_1^*, ..., \theta_d^*)^T$. Using asymptotic normality, an asymptotic $100(1-p)\%,\ 0 < p < 1$ marginal confidence interval of θ_i^* is

$$\hat{\theta}_{ni} \pm z_{1-p/2}\sqrt{\hat{\sigma}_{ii}/n}, \tag{33}$$

where $z_{1-p/2}$ is the $(1-p/2)$th quantile of a standard normal distribution $N(0,1)$. An uncorrected lower bound hyper-rectangular confidence region with at most $100(1-p)\%$ can be defined by taking d uncorrected intervals such that

$$C_{lb}(z_{p/2}) = \prod_{i=1}^{n} \left[\hat{\theta}_{ni} - z_{1-p/2}\sqrt{\hat{\sigma}_{ii}/n}, \hat{\theta}_{ni} + z_{1-p/2}\sqrt{\hat{\sigma}_{ii}/n},\right]. \tag{34}$$

Applying a Bonferroni correction gives

$$C_{ub}(z_{p/2d}) = \prod_{i=1}^{n} \left[\hat{\theta}_{ni} - z_{1-p/2d}\sqrt{\hat{\sigma}_{ii}/n}, \hat{\theta}_{ni} + z_{1-p/2d}\sqrt{\hat{\sigma}_{ii}/n}, \right], \tag{35}$$

naturally, $C_{lb}(z_{p/2}) \subseteq C_{ub}(z_{p/2d})$. With a quasi Monte-Carlo approach, one can obtain a z^* such that $z_{1-p/2} < z^* < z_{1-p/2d}$ and $C_{lb}(z_{p/2}) \subseteq C(z^*) \subseteq C_{ub}(z_{p/2d})$, with $\mathbb{P}(\theta^* \in C(z^*)) \approx 1-p$ assuming $\hat{\theta}_n \approx N_p\left(\theta^*, \hat{\Sigma}_n\right)$. A depiction of the different confidence regions can be found in figure 8. Additionally, a simple bivariate linear regression example is provided in the appendix. The corresponding implementation is provided as well.

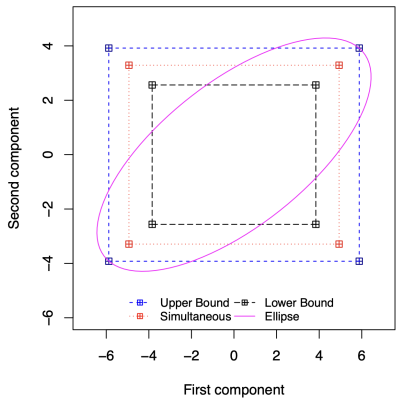

Figure 8: Singh et al. (2023). Comparison of different confidence regions.

Generally, the equal-size batching strategy is applicable to averaging over any fixed k-neighboring batches, $k \in \mathbb{N}$. However, large k will reduce the efficiency of the covariance estimate, due to the reduced number of batches. The straightforward advantages of their method are that valid frequentist inference can be conducted if only gradients are available, thereby not requiring a closed form solution or second order methods. However, the batch-means estimator still scales quadratically in the number of parameters. The theoretical framework is applicable to other SGD variants, i.e., including momentum or averaged implicit SGD, as long as $\sqrt{n}\left(\hat{\theta}_n - \theta^*\right) \xrightarrow{d} N(0, \Sigma)$, as $n \to \infty$ holds. However, since the learning rate is fixed to $\eta_i = \eta i^{-\alpha}, \alpha \in (0.5, 1), i = 1, 2, 3, ...,$ an application to adaptive learning rate algorithms like adaptive moment estimation (ADAM) (Kingma and Ba, 2014) becomes infeasible and would require the development of more extended theory.

3 Discussion

The analysis of the previous research has identified the following key challenges of DUQ in DNN: (1) Evaluation of the covariance matrix of a DNN is often computationally infeasible and thus requires meaningful approximations (Daxberger et al., 2021, Mishkin et al., 2018, Abdar et al., 2021). (2) Because there is no established theory for conducting frequentist inference on DNNs, indirect approaches like SSR inference (Dorigatti et al.,

2023) or Laplace redux (Daxberger et al., 2021) can provide approximate inference on the DNN structure. Further, Singh et al. (2023) show that for convex objectives and an ASGD optimizer, a covariance matrix can be estimated via a batch-means estimator.

The LA uses a Gaussian approximation to the posterior distribution by transforming a FNN post hoc to a BNN, using the MAP estimate and local Hessian as mean and covariance respectively. Approaches like KFAC or low rank approximations as introduced by Mishkin et al. (2018) offer a tradeoff between computational efficiency and a meaningful representation of the actual model covariance matrix. Shallow approaches, such as mean-field approximations that are commonly used in VI may be naive in the presence of strong posterior correlations and thus offer poor DUQ, since off-diagonal complexities of the covariance matrix are ignored. From a frequentist perspective, LA provides confidence intervals for the (biased) parameters learned during regularized risk minimization. Naturally, this approach is strictly dependent on the quality of the local minimum and the approximation of the covariance matrix, which may be infeasible for a large pretrained network. However, the authors also offer subnetwork inference, which is based on the assumption that a DNN can be pruned extensively without a notable loss of test accuracy (Frankle and Carbin, 2019). Moreover, last layer inference can also be conducted, but, as the name implies, does not capture the complexity of the entire parameter landscape of the DNN. Last layer inference can still be of interest in large pretrained models. In many cases, computational resources may be too constrained to even approximate a subnetwork of a large DNN and the LA generally tends to underfit (Daxberger et al., 2021). Also, it remains questionable how effectively a multivariate normal approximation like the LA performs, especially considering the highly complex nature of DNNs.

The framework introduced by Dorigatti et al. (2023) uses the theory of GLMMs to incorporate the uncertainty of a DNN into a SSR model. Here, the uncertainty of the DNN is treated as a random offset and incorporated into the estimate of the coefficients for the additive predictor, thereby enabling confidence intervals of nominal coverage and a decrease of the Type-I error rate. However, as stated by the authors, the approach is strictly dependent on the DUQ of the DNN, meaning that a poor estimation of the covariance matrix cannot be compensated for with their method. Naturally, DUQ becomes more important if the unstructured effect in the data increases. Estimates of the covariance matrix were obtained using ensembling and BBP for the regular SSR models. For the uncertainty unaware DNN baseline (covariance matrix was simply discarded) and the GLMMs, MCD and ensembling was used. Ensembling is a naturally powerful technique to obtain uncertainty quantification but also costly. As the authors stated themselves, their experiments are limited to the selected DUQ. Using approaches like SLANG may further improve the DUQ of a regular SSR due to the more complex representation of the covariance matrix. Moreover, the LA could also be an interesting choice, as one can obtain uncertainty of a FNN post hoc. Both approaches model the posterior distribution as a Gaussian. Therefore, rendering them applicable to the main theorem stated by the authors, potentially providing improved coverage. However, one may also notice that a Gaussian approximation to the posterior distribution is quite naive. The development of more flexible approaches to model the distribution of the random offset could further

improve DUQ and consequently the performed inference on the structured coefficients.

Singh et al. (2023) use the consistency of the ASGD to create an asymptotically normal and consistent estimate of the covariance matrix with a batch-means estimator, using an equal-size batching strategy for convex objective functions. This framework is particularly useful when only gradients are available, as it does not require a closed form solution or second-order methods (note that the batch-means estimator still scales quadratically in the number of parameters), and is applicable if the optimization task is convex, e.g., in linear or logistic regression. However, as stated in the appendix and by Dorigatti et al. (2023), the optimization of the additive predictor of the SSR model is convex. Furthermore, SGD is used by the authors to obtain the estimates of the structured coefficients, potentially enabling the application to the SSR framework. One may differentiate that the method of Singh et al. (2023) is not directly applicable to DNNs because of the multimodality of the loss surface. However, their approach may still improve the inference on the structured coefficients that already incorporate the uncertainty of the DNN predictions obtained by DUQ. This renders the application of marginal-friendly simultaneous confidence intervals and hypothesis testing feasible for structured coefficients obtained by ASGD, especially when a closed form solution may not be computationally feasible, due to a large number of parameters. Singh et al. (2023) state that their theoretical framework is extendable to any SGD estimator as long as the asymptotic normality of the ASGD estimator is fulfilled. Hence, rendering it applicable to averaged implicit SGD and also to SGD with momentum. However, the authors do not explicitly state that their framework can be used for adaptive learning rate algorithms like ADAM, which are commonly used for DNN, as it requires specific assumptions on the learning rate. Therefore, applying SGD inference on DNNs remains challenging, as it requires a theoretical framework that is applicable to non-convex functions and adaptive learning rate algorithms.

4 Conclusion

Performing statistical inference on DNNs is an extraordinarily difficult task and requires a tradeoff between efficient resource handling and meaningful approximations to the underlying DNN structure. The presented approaches are all based on a number of assumptions; these assumptions are necessary to constrain the complexity of the given approximation and optimization problems, because the true DNN covariance structure is generally infeasible. Therefore, future research must further improve second-order optimization and posterior approximation techniques. Moreover, the theory on FNN inference needs to be extended. Methods like SSR provide hopeful approaches, to at least indirectly, model DNN uncertainty and can be improved upon by advancing DUQ methods. Moreover, extending the SGD inference framework to adaptive learning rate algorithms and non-convex objectives would provide a powerful method to perform frequentist inference on DNNs, if the scalability of the estimation is handled appropriately and convergence can be achieved.

A Appendix

Proof of convexity for 2.2.1.

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\boldsymbol{\beta} \in \mathbb{R}^d$, $\boldsymbol{f} \in \mathbb{R}^n$ and $\boldsymbol{\epsilon} \in \mathbb{R}^n$. Denote $\mathbf{y} \in \mathbb{R}^n$ and assume the data generating process $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{f} + \boldsymbol{\epsilon}$, with $\mathbf{z} \in \mathbb{R}^n$, $\mathbf{z} \sim \mathcal{N}(\boldsymbol{f}, \boldsymbol{\Gamma})$, and known full-rank covariance matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{n \times n}$. Further, let $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. Using the mean squared error, one obtains the empirical risk

$$\mathcal{R}_{emp}(\boldsymbol{\beta}) = ||(\mathbf{y} - \mathbf{z}) - \mathbf{X}\boldsymbol{\beta}||^2.$$

Taking the first and second derivative with respect to $\boldsymbol{\beta}$ yields

$$\frac{\partial}{\partial \boldsymbol{\beta}} ||(\mathbf{y} - \mathbf{z}) - \mathbf{X}\boldsymbol{\beta}||^2 = 2\left(-\mathbf{X}^T(\mathbf{y} - \mathbf{z}) + \mathbf{X}^T\mathbf{X}\boldsymbol{\beta}\right),$$

$$\frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} ||(\mathbf{y} - \mathbf{z}) - \mathbf{X}\boldsymbol{\beta}||^2 = 2\mathbf{X}^T\mathbf{X}.$$

Because for any $v \in \mathbb{R}^d, v \neq 0$ it holds that $v^T\mathbf{X}^T\mathbf{X}v = (\mathbf{X}v)^T(\mathbf{X}v) = ||\mathbf{X}v||^2 \geq 0 \Leftrightarrow 2||\mathbf{X}v||^2 \geq 0$, the objective function is convex.

Proof of convexity for 2.2.2.

Assuming the same data generating process as before, the penalized empirical risk problem with mean-squared error and an L_2 constraint for the coefficients gives

$$\mathcal{R}_{reg}(\boldsymbol{\beta}) = ||(\mathbf{y} - \mathbf{z}) - \mathbf{X}\boldsymbol{\beta})||^2 + \mathbf{S}_\lambda ||\boldsymbol{\beta}||^2.$$

Taking derivatives

$$\frac{\partial}{\partial \boldsymbol{\beta}} (||(\mathbf{y} - \mathbf{z}) - \mathbf{X}\boldsymbol{\beta})||^2 + \mathbf{S}_\lambda ||\boldsymbol{\beta}||) = 2\left(-\mathbf{X}^T(\mathbf{y} - \mathbf{z}) + \mathbf{X}^T\mathbf{X}\boldsymbol{\beta} + \mathbf{S}_\lambda\boldsymbol{\beta}\right),$$

$$\frac{\partial^2}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} ||(\mathbf{y} - \mathbf{z}) - \mathbf{X}\boldsymbol{\beta})||^2 = 2\mathbf{X}^T\mathbf{X} + 2\mathbf{S}_\lambda.$$

For any $v \in \mathbb{R}^d, v \neq 0$

$$v^T\left(\mathbf{X}^T\mathbf{X} + \mathbf{S}_\lambda\right)v = \underbrace{||\mathbf{X}v||^2}_{\geq 0} + v^T\mathbf{S}_\lambda v \geq 0$$

$$\Leftrightarrow ||\mathbf{X}v||^2 \geq -v^T\mathbf{S}_\lambda v.$$

Therefore, for $v^T\mathbf{S}_\lambda v \geq -||\mathbf{X}v||^2$, the objective function is convex. It is straightforward to see that if $\mathbf{S}_\lambda$ is at least positive semi-definite, the resulting objective is convex, e.g., in Ridge regression.

ASGD Example.

Assume the relationship $y = 2x_1 + 0.5x_2 + \epsilon$, with independent $x_1, x_2 \sim U(-5, 5)$ and $\epsilon \sim N(0, 0.1^2)$. Define the loss function as $f : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}, f(\theta, \zeta) = \frac{1}{2}\left(y - x^T\theta\right)^2$, where the expected loss is given by $F(\theta) = \mathbb{E}_\Pi\left[\frac{1}{2}\left(y - x^T\theta\right)^2\right]$, with $x, \theta \in \mathbb{R}^2$ and $y \in \mathbb{R}$. Recall that $\mathbb{E}[x^2] = Var(x) + (\mathbb{E}[x])^2$. For $x \sim U(a, b)$, it hols that $\mathbb{E}[x] = \frac{1}{2}(a + b)$ and $Var(x) = \frac{1}{12}(b - a)^2$. Also, observe that $\mathbb{E}_\Pi[x_1x_2] = \mathbb{E}_\Pi[x_1]\mathbb{E}_\Pi[x_2]$ and $\mathbb{E}_\Pi[x_1] = \mathbb{E}_\Pi[x_2] = \mathbb{E}_\Pi[\epsilon] = 0$. The true minimizer is then given by

$$\begin{aligned}
\arg\min_{\theta\in\mathbb{R}^2} F(\theta) &= \arg\min_{\theta\in\mathbb{R}^2} \mathbb{E}_\Pi\left[\frac{1}{2}(y - x^T\theta)^2\right] \\
&= \arg\min_{\theta\in\mathbb{R}^2} \frac{1}{2}\mathbb{E}_\Pi\left[y^2 - 2yx^T\theta + \theta^T xx^T\theta\right] \\
&= \arg\min_{\theta\in\mathbb{R}^2} \frac{1}{2}\left(\mathbb{E}_\Pi\left[y^2\right] - 2\mathbb{E}_\Pi\left[yx^T\right]\theta + \theta^T\mathbb{E}_\Pi\left[xx^T\right]\theta\right) \\
&= -\mathbb{E}_\Pi[yx] + \mathbb{E}_\Pi\left[xx^T\right]\theta.
\end{aligned}$$

Setting $-\mathbb{E}_\Pi[yx] + \mathbb{E}_\Pi\left[xx^T\right]\theta = 0$ yields

$$\begin{aligned}
\mathbb{E}_\Pi[yx] &= \mathbb{E}_\Pi\left[xx^T\right]\theta \\
\Leftrightarrow \mathbb{E}_\Pi\left[\begin{pmatrix} 2x_1^2 + 0.5x_1x_2 + \epsilon x_1 \\ 2x_1 + 0.5x_2^2 + \epsilon x_2 \end{pmatrix}\right] &= \mathbb{E}_\Pi\left[\begin{pmatrix} x_1^2 & x_1x_2 \\ x_1x_2 & x_2^2 \end{pmatrix}\right]\theta \\
\Leftrightarrow \begin{pmatrix} 2\mathbb{E}_\Pi\left[x_1^2\right] + 0.5\mathbb{E}_\Pi[x_1x_2] + \mathbb{E}_\Pi[\epsilon x_1] \\ 2\mathbb{E}_\Pi[x_1x_2] + 0.5\mathbb{E}_\Pi\left[x_2^2\right] + \mathbb{E}_\Pi[\epsilon x_2] \end{pmatrix} &= \begin{pmatrix} \mathbb{E}_\Pi\left[x_1^2\right] & \mathbb{E}_\Pi[x_1x_2] \\ \mathbb{E}_\Pi[x_1x_2] & \mathbb{E}_\Pi\left[x_2^2\right] \end{pmatrix}\theta \\
\Leftrightarrow \begin{pmatrix} 2\mathbb{E}_\Pi\left[x_1^2\right] \\ 0.5\mathbb{E}_\Pi\left[x_2^2\right] \end{pmatrix} &= \begin{pmatrix} \mathbb{E}_\Pi\left[x_1^2\right] & 0 \\ 0 & \mathbb{E}_\Pi\left[x_2^2\right] \end{pmatrix}\theta \\
\theta &= \begin{pmatrix} \mathbb{E}_\Pi\left[x_1^2\right] & 0 \\ 0 & \mathbb{E}_\Pi\left[x_2^2\right] \end{pmatrix}^{-1}\begin{pmatrix} 2\mathbb{E}_\Pi\left[x_1^2\right] \\ 0.5\mathbb{E}_\Pi\left[x_2^2\right] \end{pmatrix} \\
\theta^* &= \begin{pmatrix} 2 \\ 0.5 \end{pmatrix}.
\end{aligned}$$

It can be shown that this is indeed the minimizer by proving the convexity of $F(\theta)$.

$$\begin{aligned}
\nabla^2 F(\theta) &= \nabla_\theta\left(-\mathbb{E}_\Pi[yx] + \mathbb{E}_\Pi\left[xx^T\right]\theta\right) \\
&= \mathbb{E}_\Pi\left[xx^T\right] \\
&= \begin{pmatrix} \mathbb{E}_\Pi\left[x_1^2\right] & \mathbb{E}_\Pi[x_1x_2] \\ \mathbb{E}_\Pi[x_1x_2] & \mathbb{E}_\Pi\left[x_2^2\right] \end{pmatrix} \\
&= \begin{pmatrix} Var_\Pi(x_1) + (\mathbb{E}_\Pi[x_1])^2 & 0 \\ 0 & Var_\Pi(x_2) + (\mathbb{E}_\Pi[x_2])^2 \end{pmatrix} \\
&= \begin{pmatrix} Var_\Pi(x_1) & 0 \\ 0 & Var_\Pi(x_2) \end{pmatrix} \\
&= A.
\end{aligned}$$

Because the eigenvalues of $\nabla^2 F(\theta)$ are $\lambda_1 = \lambda_2 = Var_\Pi(x_1) = Var_\Pi(x_2) > 0$, $F(\theta)$ is strictly convex and θ^* is the unique minimizer of $F(\theta)$. Further, note that $\nabla^2 F(\theta) = \nabla^2 F(\theta^*)$ because F is a quadratic function with a unique minimum.

To finally derive Σ, it is required to obtain $S = \mathbb{E}_\Pi\left(\left[\nabla f(\theta^*, \zeta)\right]\left[\nabla f(\theta^*, \zeta)\right]^T\right)$. Since $f(\theta, \zeta) = \frac{1}{2}\left(y - x^T\theta\right)^2$ and $\nabla f(\theta, \zeta) = -\left(y - x^T\theta\right)$, it follows that

$$\begin{aligned}\nabla f(\theta^*, \zeta) &= -\left(y - x^T\theta^*\right)x \\ &= -(2x_1 + 0.5x_2 + \epsilon - (2x_1 + 0.5x_2))x \\ &= -\epsilon x.\end{aligned}$$

Consequently,

$$\begin{aligned}S &= \mathbb{E}_\Pi\left(\left[\nabla f(\theta^*, \zeta)\right]\left[\nabla f(\theta^*, \zeta)\right]^T\right) \\ &= \mathbb{E}_\Pi\left([-\epsilon x][-\epsilon x]^T\right) \\ &= \mathbb{E}_\Pi\left[\epsilon^2\right]\mathbb{E}_\Pi\left[xx^T\right] \\ &= Var_\Pi(\epsilon)A.\end{aligned}$$

Since $\Sigma = A^{-1}SA^{-1}$ and $S = Var_\Pi(\epsilon)A$,

$$\begin{aligned}\Sigma &= Var_\Pi(\epsilon)A^{-1}AA^{-1} \\ &= Var_\Pi(\epsilon)A^{-1}.\end{aligned}$$

For $x_1, x_2 \sim U(-5, 5)$ and $\epsilon \sim N(0, 0.1^2)$, this results in

$$\Sigma = 0.1^2\begin{pmatrix}12/100 & 0 \\ 0 & 12/100\end{pmatrix}.$$

B Electronic Appendix

The implementation of the ASGD example is provided in the notebook `sgd_inference_implementation.ipynb`.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297.

Barber, R., Candès, E., Ramdas, A., and Tibshirani, R. (2021). Predictive inference with the jackknife+. *Annals of Statistics*, 49:486–507.

Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Thomson Learning, 2 edition.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. (2021). Laplace redux — effortless bayesian deep learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, pages 20089–20103. Curran Associates, Inc.

Dorigatti, E., Schubert, B., Bischl, B., and Ruegamer, D. (2023). Frequentist uncertainty quantification in semi-structured neural networks. In Ruiz, F., Dy, J., and van de Meent, J.-W., editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 1924–1941. PMLR.

Frankle, J. and Carbin, M. (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Goan, E. and Fookes, C. (2020). *Bayesian Neural Networks: An Introduction and Survey*, page 45–87. Springer International Publishing.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Lehmann, E. L. and Romano, J. P. (2007). *Testing Statistical Hypotheses*. 3 edition.

Liu, H., Ong, Y.-S., Shen, X., and Cai, J. (2020). When gaussian process meets big data: a review of scalable gps. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4405–4423.

Maddox, W. J., Benton, G. W., and Wilson, A. G. (2020). Rethinking parameter counting in deep models: Effective dimensionality revisited. *CoRR*, abs/2003.02139.

Mishkin, A., Kunstner, F., Nielsen, D., Schmidt, M. W., and Khan, M. E. (2018). Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient. *ArXiv*, abs/1811.04504.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press.

Rügamer, D., Kolb, C., and Klein, N. (2023). Semi-structured distributional regression – extending structured additive models by arbitrary deep neural networks and data modalities. *The American Statistician*, 0(ja):1–25.

Singh, R., Shukla, A., and Vats, D. (2023). A workflow for statistical inference in stochastic gradient descent. *arXiv preprint arXiv:2303.07706*.

Tian, Y., Zhang, Y., and Zhang, H. (2023). Recent advances in stochastic gradient descent in deep learning. *Mathematics*, 11(3).

Vats, D. and Flegal, J. M. (2022). Lugsail lag windows for estimating time-average covariance matrices. *Biometrika*, 109(3):735–750.

Declaration of authorship

I hereby declare that the report submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the Thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the report as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future Theses submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.

Munich, February 6$^{\text{th}}$, 2024

Name