Sampling and response noise epistemic uncertainties: from linear regressors to linearized deep networks

Pierre Nodet
Orange Research
Châtillon, France
pierre.nodet@orange.com

Thomas George
Orange Research
Châtillon, France
thomas.george@orange.com

Abstract

This work investigates the impact of two distinct sources of randomness on the uncertainty associated with machine learning models trained on finite samples. By extending estimators derived in linear regression to non-linear (deep learning) hypothesis classes, we quantify how these sources of uncertainty differentially influence predictions at individual test points within the instance space. Our analysis underscores the role of the hypothesis class in shaping the prediction's uncertainty landscape, which is illustrated in 2 experimental setups.

1 Introduction

For AI systems deployed in real-world applications, quantifying uncertainty in a given prediction is key, for example, to fall back to human monitoring when uncertain in high-stakes contexts. In supervised machine learning, we suppose a true underlying model $p(x, y)$ on an instance space $\mathcal{X}$ and response space $\mathcal{Y}$, that we observe through a finite training dataset $\mathcal{D} := \{(\boldsymbol{x}_i, y_i)\}_{1 \leq i \leq n}$, and we want to estimate $p(y|x_{\text{test}})$ for future test points x_{test}. Here, uncertainty comes both from the uncertainty $p(y|x)$ of the process that we are trying to estimate (termed aleatoric uncertainty), as well as uncertainty resulting from our particular finite-sample knowledge of data from this process (epistemic uncertainty); see [7] for a complete introduction.

We focus on the latter, we take a closer look at epistemic uncertainty by identifying contributions from 2 different sources to the variance of the prediction for fixed test points: the training dataset is a finite sample of the true generating process, which partially captures the structure of the training set: instances $\{\boldsymbol{x}_i\}_{1 \leq i \leq n}$ only partially cover the instance space, and responses $\{y_i\}_{1 \leq i \leq n}$ are a single realization of a noisy process. A machine learning algorithm trained on this dataset inherits this limited knowledge, which translates into uncertainty in future test predictions. Ultimately, we would like to be able to answer the question: *do I need to gather more training points or do I need to review labels more carefully in order to most effectively reduce epistemic uncertainty?*

Our perspective is inspired by insights from ordinary least squares, which are well-studied and have offered many theoretical results in statistics, that we apply to deep learning using the (empirical) tangent kernel/features framework.

Contributions and organization of the document After reviewing some classical results from parametric statistics (sections 2.1 and 2.2), we propose to directly adapt these estimators to linearized deep networks (section 2.3). On illustrative experiments (sections 3 and 4), we show that these estimators offer new insights into the effect on test predictions of two sources of epistemic uncertainty.

1st EurIPS Workshop on Epistemic Intelligence in Machine Learning (EIML 2025) .

2 Background

2.1 Linear regression: dependence on finite samples and resulting heteroscedasticity

We start with the simpler setup of linear regression, where we observe a linear data generating process $\mathrm{y} = \boldsymbol{w}^{*\top}\mathbf{x} + \varepsilon$, with $\mathbf{x} \sim \mathcal{N}(0, \Sigma)$ and independent noise $\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$, through a sample of n observations $\{(\boldsymbol{x}_i, y_i)\}_{1 \leq i \leq n}$ (the training dataset), that we stack in the $n \times d$ design matrix X and the n response vector $\boldsymbol{y} = X\boldsymbol{w}^* + \boldsymbol{\varepsilon}$.

As a classical textbook result of the ordinary least squares (OLS) problem, we obtain a closed-form solution to the estimator of the prediction for a test point x_{test} assuming that $X^\top X$ has full rank:

$$\hat{y}(\boldsymbol{x}_{\text{test}}, X, \boldsymbol{\varepsilon}) = \boldsymbol{x}_{\text{test}}^\top \left(X^\top X\right)^{-1} X^\top (X\boldsymbol{w}^* + \boldsymbol{\varepsilon}) \tag{1}$$

$$= \boldsymbol{x}_{\text{test}}^\top \boldsymbol{w}^* + \underbrace{\boldsymbol{x}_{\text{test}}^\top \left(X^\top X\right)^{-1} X^\top \boldsymbol{\varepsilon}}_{\text{depends on } X, \boldsymbol{\varepsilon}} \tag{2}$$

This highlights the dependence of the estimator on the particular samples X and $\boldsymbol{\varepsilon}$. Different samples would provide different estimators, and the link between $(X, \boldsymbol{\varepsilon})$ and the estimator $\hat{y}(\boldsymbol{x}_{\text{test}}, X, \boldsymbol{\varepsilon})$ is constrained by our particular choice of hypothesis class: here, linear regressors.

Uncertainty and variance We study the epistemic uncertainty that stems from this dependence on finite samples: the estimator $\hat{y}(\boldsymbol{x}_{\text{test}}, \mathbf{X}, \boldsymbol{\varepsilon})$ is a random variable that depends on random variables $\mathbf{X}$ and $\boldsymbol{\varepsilon}$, thus a particular estimate with a given sample is just a realization of this random variable. We aim to calculate the variance of this estimator as a measure of uncertainty.

Variance due to response noise Here the set of instances X is considered fixed, and the response noise $\boldsymbol{\varepsilon}$ is a random variable. The variance of the estimated prediction

$$\text{var}\left(\hat{y}(\boldsymbol{x}_{\text{test}}, X, \boldsymbol{\varepsilon}) | X\right) = \text{var}\left(\boldsymbol{x}_{\text{test}}^\top \left(X^\top X\right)^{-1} X^\top \boldsymbol{\varepsilon} | X\right) \tag{3}$$

involves an expectation over the random variable ε. It highlights the role of the chosen hypothesis class in the dependence on aleatoric uncertainty.

Variance due to sampling $(X, \boldsymbol{\varepsilon})$ A second source of variance is attributable to the particular sample of covariates X and its response noise $\boldsymbol{\varepsilon}$: had we gathered different training examples and their respective labels, we would have obtained a different estimator. This is quantified as the total variance

$$\text{var}\left(\hat{y}(\boldsymbol{x}_{\text{test}}, X, \boldsymbol{\varepsilon})\right) = \text{var}\left(\boldsymbol{x}_{\text{test}}^\top \left(X^\top X\right)^{-1} X^\top \boldsymbol{\varepsilon}\right) \tag{4}$$

which involves expectations over both random variables.

Heteroscedasticity Interestingly, both variance terms depend on the choice of test point $\boldsymbol{x}_{\text{test}}$. Even in a setup with homoscedastic noise in the data generating process, the modeling choice of hypothesis class shapes how uncertainty affects different test points.

2.2 Variance estimators for linear regression

In practice, we do not have access to the true data generating process. Our goal is thus to provide estimators of both variance terms using only quantities computed using a given sample of n examples.

Variance due to response noise For fixed X, the variance of the estimated prediction admits a closed form expression:

$$\text{var}\left(\hat{y}(\boldsymbol{x}_{\text{test}}, X, \boldsymbol{\varepsilon}) | X\right) = \sigma^2 \boldsymbol{x}_{\text{test}}^\top \left(X^\top X\right)^{-1} \boldsymbol{x}_{\text{test}} \tag{5}$$

When the true noise variance is unknown, we use the estimator $\hat{\sigma}^2 = \frac{1}{n-d}\sum_{i=1}^{n}\left(y_i - \hat{y}(\boldsymbol{x}_i, X, \boldsymbol{\varepsilon})\right)^2$, which gives the OLS estimator for the uncertainty stemming from response noise:

$$\widehat{\text{var}}_{\text{OLS}}\left(\hat{y}(\boldsymbol{x}_{\text{test}}, X, \boldsymbol{\varepsilon}) | X\right) = \hat{\sigma}^2 \boldsymbol{x}_{\text{test}}^\top \left(X^\top X\right)^{-1} \boldsymbol{x}_{\text{test}} \tag{6}$$

Variance due to sampling Similarly, we want to estimate the variance of the estimator on the finite-sample $(X, \boldsymbol{\varepsilon})$. Closed-form expressions, even in simplified settings with normal distributions, involve estimating moments from Wishart laws [e.g. in 3]. For simplicity, we instead resort to jackknife estimates [16, 19]. As a first step, we seek an estimator of the variance of the estimated parameters $\hat{\boldsymbol{w}}$. $\hat{\boldsymbol{w}}^{(-i)}$ denotes the parameters obtained when example i is left out of the training data.

$$\hat{\text{var}}_{\text{jackknife}}(\hat{\boldsymbol{w}}) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \left(\bar{\boldsymbol{w}} - \bar{\boldsymbol{w}}^{(-i)} \right)^2$$

where $\bar{\boldsymbol{w}}^{(-i)} = n\hat{\boldsymbol{w}} - (n-1)\hat{\boldsymbol{w}}^{(-i)}$ and $\bar{\boldsymbol{w}} = \frac{1}{n} \sum_{i=1}^{n} \bar{\boldsymbol{w}}^{(-i)}$

Yet the jackknife estimate of the parameters $\bar{\boldsymbol{w}}$ is different from the OLS estimate $\hat{\boldsymbol{w}}$ [6]. We instead propose to use an alternative definition from [11] which assumes that $\bar{\boldsymbol{w}} \approx \hat{\boldsymbol{w}}$:

$$\hat{\text{var}}_{\text{jackknife}}(\hat{\boldsymbol{w}}) = \frac{n-1}{n} \sum_{i=1}^{n} \left(\hat{\boldsymbol{w}} - \hat{\boldsymbol{w}}^{(-i)} \right)^2 \tag{7}$$

This definition has the advantage of proposing a simple closed-form formula for linear regression using the Sherman-Morrison formula [18]:

$$\hat{\boldsymbol{w}}^{(-i)} = \hat{\boldsymbol{w}} - \frac{1}{1-h_{ii}} \left(X^\top X \right)^{-1} \boldsymbol{x}_i \hat{e}_i$$

where $h_{ii} = \boldsymbol{x}_i^\top \left(X^\top X \right)^{-1} \boldsymbol{x}_i$ is the leverage of $\boldsymbol{x}_i$ and $\hat{e}_i = y_i - \hat{y}(\boldsymbol{x}_i)$ is the residual of the model $\hat{y}$ at $\boldsymbol{x}_i$.

Substituting in Equation 7, we obtain:

$$\hat{\text{var}}_{\text{jackknife}}(\hat{\boldsymbol{w}}) = \frac{n-1}{n} \left(X^\top X \right)^{-1} \sum_{i=1}^{n} \left(\frac{\hat{e}_i^2}{(1-h_{ii})^2} \boldsymbol{x}_i^\top \boldsymbol{x}_i \right) \left(X^\top X \right)^{-1}$$

The jackknife estimate of the variance of the parameters is robust to both heteroscedastic noise and outliers (high-leverage examples) [10]. However, contrary to the OLS estimator of the variance of the parameters, this estimator is not unbiased [6, 10].

Finally, we can use this estimator of the variance of the parameters to quantify the uncertainty at a test point $\boldsymbol{x}_{\text{test}}$ that stems from sampling $(X, \boldsymbol{\varepsilon})$:

$$\hat{\text{var}}_{\text{jackknife}} \left(\hat{y} \left(\boldsymbol{x}_{\text{test}}, X, \boldsymbol{\varepsilon} \right) \right) = \boldsymbol{x}_{\text{test}}^\top \left(X^\top X \right)^{-1} \frac{n-1}{n} \sum_{i=1}^{n} \left(\frac{\hat{e}_i^2}{(1-h_{ii})^2} \boldsymbol{x}_i^\top \boldsymbol{x}_i \right) \left(X^\top X \right)^{-1} \boldsymbol{x}_{\text{test}} \tag{8}$$

2.3 Variance estimators in deep learning

We now turn to non-linear data generating processes in the form $\mathrm{y} = f_{\boldsymbol{w}^*}(\mathbf{x}) + \varepsilon$, where the functional $\boldsymbol{w} \mapsto f_{\boldsymbol{w}}$ is non-linear. This setup comprises neural networks, where parameters $\boldsymbol{w}$ are weights and biases of all layers, arranged as a d vector. Given a training dataset $\mathcal{D} := \{(\boldsymbol{x}_i, y_i)\}_{1 \leq i \leq n}$ of observations from this process, we want to conduct maximum likelihood estimation (MLE) of the parameters $\boldsymbol{w}$ by minimizing the negative log-likelihood (NLL):

$$-\log \mathcal{L}(\mathcal{D}, \boldsymbol{w}) = \sum_{i=1}^{n} \ell(f_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)$$

where $\ell(f_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i) = (y_i - f_{\boldsymbol{w}}(\boldsymbol{x}_i))^2$ for independent gaussian noise ε. Contrary to OLS linear regression, MLE in deep learning does not admit a closed-form solution and needs to be solved by iterative algorithms like stochastic gradient descent [SGD 17]. We suppose that we have access to the minimizer of the NLL as our estimator on parameters, which we denote $\hat{\boldsymbol{w}}$. In order to derive estimators of both variance terms Equation 3 and 4, we will linearize the functional $\boldsymbol{w} \mapsto f_{\boldsymbol{w}}$ in $\boldsymbol{w} = \hat{\boldsymbol{w}}$:

$$f_{\boldsymbol{w}}(\cdot) = f_{\hat{\boldsymbol{w}}}(\cdot) + \delta \boldsymbol{w}^\top \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\cdot) + \text{H.O.T.} \tag{9}$$

where $\phi_{\hat{\boldsymbol{w}}}(\cdot) = \left.\frac{\partial f_{\boldsymbol{w}}(\cdot)}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\hat{\boldsymbol{w}}}$ are called the tangent features [8, 2]. During training of a neural network, it has been empirically shown [5] that after a short initial phase of representation learning where tangent features rotated and stretched to adapt to the particular task learned [1], training stabilized in linearly connected modes where the linearization in Equation 9 essentially captured the actual training dynamics (higher orders vanish). We thus build our estimators by leveraging the analogy with linear models applied on top of tangent features $\phi_{\hat{\boldsymbol{w}}}(\cdot)$, considered fixed in the vicinity of $\hat{\boldsymbol{w}}$, used as an anchor. We make the assumption that orders greater than 1 are negligible when considering different samples X and $\boldsymbol{\varepsilon}$.

For training examples X and noise vector $\boldsymbol{\varepsilon}$, we recover the OLS problem with linearized predictor in order to obtain an estimator for $\delta\boldsymbol{w}$:

$$\left\|f_{\boldsymbol{w}^*}(X) + \boldsymbol{\varepsilon} - f_{\boldsymbol{w}}(X)\right\|^2 \approx \left\| \underbrace{f_{\boldsymbol{w}^*}(X) - f_{\hat{\boldsymbol{w}}}(X) + \boldsymbol{\varepsilon}}_{:= \text{ pseudo-responses } \boldsymbol{r}} - \Phi_{\boldsymbol{w}} \delta \boldsymbol{w} \right\|^2 \tag{10}$$

where $\Phi_{\boldsymbol{w}} := \begin{pmatrix} - & \boldsymbol{\phi}_{\boldsymbol{w}}(\boldsymbol{x}_1)^\top & - \\ & \vdots & \\ - & \boldsymbol{\phi}_{\boldsymbol{w}}(\boldsymbol{x}_n)^\top & - \end{pmatrix}$ are the $n \times d$ stacked tangent features, and $f_{\boldsymbol{w}}(X) := (f_{\boldsymbol{w}}(\boldsymbol{x}_1), \ldots, f_{\boldsymbol{w}}(\boldsymbol{x}_n))^\top$ denotes $f_{\boldsymbol{w}}$ applied to every row of X. This OLS problem for estimating $\delta\boldsymbol{w}$ involves pseudo-responses $\boldsymbol{r}$ and transformed design matrix $\Phi_{\boldsymbol{w}}$. It admits the closed-form solution:

$$\hat{y}(\boldsymbol{x}_{\text{test}}, X, \boldsymbol{\varepsilon}) = f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_{\text{test}}) + \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_{\text{test}})^\top \left(\Phi_{\hat{\boldsymbol{w}}}^\top \Phi_{\hat{\boldsymbol{w}}}\right)^{-1} \Phi_{\hat{\boldsymbol{w}}}^\top \left(f_{\boldsymbol{w}^*}(X) - f_{\hat{\boldsymbol{w}}}(X) + \boldsymbol{\varepsilon}\right) \tag{11}$$

Variance due to response noise We can directly apply the linear regression estimator of Equation 3, we obtain:

$$\text{var}\left(\hat{y}(\boldsymbol{x}_{\text{test}}, X, \boldsymbol{\varepsilon}) \,|\, X\right) = \sigma^2 \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_{\text{test}})^\top \left(\Phi_{\hat{\boldsymbol{w}}}^\top \Phi_{\hat{\boldsymbol{w}}}\right)^{-1} \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_{\text{test}}) \tag{12}$$

and we estimate σ with $\hat{\sigma}^2 = \frac{1}{n-q}\sum_{i=1}^n \left(y_i - \hat{y}(\boldsymbol{x}_i, X, \boldsymbol{\varepsilon})\right)^2$ where q is the number of effective parameters of the model $q = \sum_{i=1}^n h_{ii}$ (the relation $q = d$ should hold for a full rank $\Phi_{\hat{\boldsymbol{w}}}$ [12]), and $h_{ii} = \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_i)^\top \left(\Phi_{\hat{\boldsymbol{w}}}^\top \Phi_{\hat{\boldsymbol{w}}}\right)^{-1} \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_i)$ is the leverage of $\boldsymbol{x}_i$ by analogy with the linear leverage.

$$\hat{\text{var}}_{\text{MLE}}(f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_{\text{test}})) = \hat{\sigma}^2 \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_{\text{test}})^\top \left(\Phi_{\hat{\boldsymbol{w}}}^\top \Phi_{\hat{\boldsymbol{w}}}\right)^{-1} \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_{\text{test}}) \tag{13}$$

This estimator can be alternatively derived by observing that $\sigma^2 \left(\Phi_{\hat{\boldsymbol{w}}}^\top \Phi_{\hat{\boldsymbol{w}}}\right)^{-1}$ is the Fisher Information, quantifying the variance of the maximum likelihood parameter estimator, and using the delta method [4] to get a variance on the prediction.

Variance due to sampling Similarly, we directly apply the estimator derived in linear regression (Equation 4):

$$\hat{\text{var}}_{\text{jackknife}}(f_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_{\text{test}})) = \frac{n-1}{n} \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_{\text{test}})^\top \left(\Phi_{\hat{\boldsymbol{w}}}^\top \Phi_{\hat{\boldsymbol{w}}}\right)^{-1} \sum_{i=1}^n \frac{\hat{e}_i^2 \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_i)}{(1-h_{ii})^2} \left(\Phi_{\hat{\boldsymbol{w}}}^\top \Phi_{\hat{\boldsymbol{w}}}\right)^{-1} \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_{\text{test}}) \tag{14}$$

where $h_{ii} = \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_i)^\top \left(\Phi_{\hat{\boldsymbol{w}}}^\top \Phi_{\hat{\boldsymbol{w}}}\right)^{-1} \boldsymbol{\phi}_{\hat{\boldsymbol{w}}}(\boldsymbol{x}_i)$ is the leverage of $\boldsymbol{x}_i$ and $\hat{e}_i = y_i - \hat{y}(\boldsymbol{x}_i)$ is the residual.

Equations 13 and 14 serve at our estimators of uncertainty that stems from response noise and finite sampling of $(X, \boldsymbol{\varepsilon})$, respectively, which we now apply in non-linear experiments.

3 Experiment: synthetic data

We conduct illustrative experiments on a toy nonlinear regression task $y_i = \frac{\sin(x_i)}{x_i} + \varepsilon_i$ with 3 distinct scenarios:

1. normally distributed predictor $\mathrm{x}_i \sim \mathcal{N}(0,3)$ and homoscedastic response noise $\varepsilon_i \sim \mathcal{N}(0,0.1)$;
2. normally distributed predictor $\mathrm{x}_i \sim \mathcal{N}(0,3)$ and heteroscedastic response noise $\varepsilon_i \sim \mathcal{N}(0,\sigma_i^2/x_i^2)$ such that label noise is more present around 0 (yet with $\frac{1}{n}\sum_{i=1}^{n}\sigma_i^2 = 0.1$ average noise equal to the first scenario);
3. non-normally distributed predictor $\mathrm{u}_i \sim \mathcal{N}(0,3)$ and $x_i = 3\frac{u_i}{|u_i|} - u_i$ such that data points are rarer around 0 and homoscedastic response noise $\varepsilon_i \sim \mathcal{N}(0,0.1)$.

We sample 500 training points in scenario. 5 random white noise dimensions are added to the input space $\mathcal{X}$ to render the task more difficult. A one hidden layer perceptron with 100 neurons and $\tanh$ activation is trained with L-BFGS [13, 9], full-batch, and ℓ_2 regularization for 1000 iterations with a step size of 0.1.

The estimated variance $\hat{\mathrm{var}}_{\mathrm{MLE}}$ (Equation 13) and $\hat{\mathrm{var}}_{\mathrm{jackknife}}$ (Equation 14) are computed on the fully trained MLP and averaged over 20 runs. The true variance from $\boldsymbol{\varepsilon}$ is estimated by fixing the training predictors X and resampling the noise $\boldsymbol{\varepsilon}$, retraining the MLP, and computing its variance on all test points over 100 runs. For the variance from $\mathbf{X}$, we resample both training points $\mathbf{X}$ and noise $\boldsymbol{\varepsilon}$, retrain the MLP, and compute its variance on all test points over 100 runs.

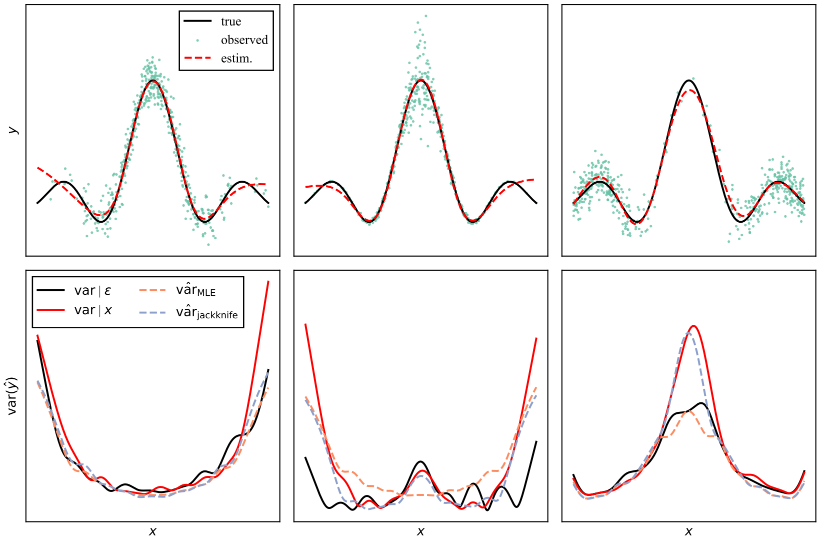

Figure 1: Estimated and true variance of an MLP over a toy nonlinear regression task.

Figure 1 shows that $\hat{\mathrm{var}}_{\mathrm{jackknife}}$ does not only explain the variance induced from sampling a given X but actually the total variance from both $\mathbf{X}$ and noise $\boldsymbol{\varepsilon}$ in all scenarios, even when the model hypotheses are not valid. $\hat{\mathrm{var}}_{\mathrm{MLE}}$ on the other hand can nicely explain the uncertainty of the model with respect to response noise, but only when the model hypotheses are valid (homoscedastic noise).

4 Experiment: California housing dataset

In addition, we qualitatively evaluate our proposed estimators (Equations 13 and 14) on the California Housing dataset [14]. These experiments serve to examine the examples for which the $\hat{\mathrm{var}}_{\mathrm{jackknife}}$ and $\hat{\mathrm{var}}_{\mathrm{MLE}}$ disagree.

As a preprocessing step, the dataset is first split into a training and a test set, and predictors are centered and then standardized. We train a one-hidden layer perceptron with 100 neurons and $\tanh$ activation with L-BFGS [13, 9], full-batch and ℓ_2 regularization for 1000 iterations with a step size of 0.1. The MLP got a MSE of 0.21 on the train set and 0.28 on the test set (the null model obtained a MSE of 1.35 on the test set).

We compare the ranking of the most uncertain test samples from $\mathbf{X}$ and ε in Figure 2. The model is less certain about its predictions (ε uncertainty) in regions with few districts or in regions with only poor districts. In these regions, the predictions of the model would be greatly affected by response noise. On the contrary, in dense regions, but with districts with high value disparities, the model uncertainty is attributable to the intrinsic difficulty of the task ($\mathbf{X}$ uncertainty).

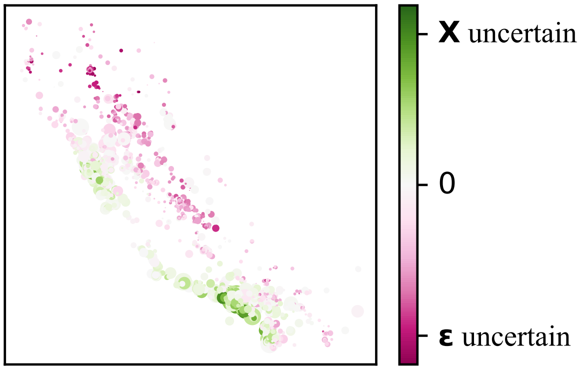

Figure 2: Map of test districts of the California Housing dataset (x-axis and y-axis correspond to longitude and latitude of the district). The circle size corresponds to the absolute errors of the MLP's predictions for every district. Districts are colored in green if the model is uncertain because of $\mathbf{X}$ or in pink because of ε.

For green regions, we would recommend acquiring more training data to help the model better distinguish valuable districts in high density regions, whereas for pink regions, data seems sufficient but careful review of label is recommended, as wrongly evaluated districts would greatly impact the model.

5 Conclusion

In this work, we highlighted the effect of two different sources of randomness on the uncertainty of machine learning models trained from finite samples, by adapting linear regression estimators to a nonlinear setup (deep learning). These estimators quantify how uncertainty from these 2 different sources differently affects each test point $\boldsymbol{x}_{\text{test}}$ of the instance space, which is a direct consequence of the modeling choice of the hypotheses class.

This preliminary work calls for further study along the following lines: First, we aim at more carefully studying the practical inaccuracies that result from the assumptions made along the way, and in particular, we should more carefully estimate the quality of linearized estimators. Second, we would like to further break down the variance due to sampling, in order to isolate the effect of X and that of the noise, whereas here the jackknife estimator accounts for both effects simultaneously (both the instance and the response are left out in jackknife estimates). Finally, we plan to scale the experiments to more complicated tasks and deeper architectures by leveraging efficient inverse Hessian vector products (iHVP) [20], as well as derive equivalent estimators in classification settings [15].

References

[1] Aristide Baratin, Thomas George, César Laurent, R Devon Hjelm, Guillaume Lajoie, Pascal Vincent, and Simon Lacoste-Julien. Implicit regularization via neural feature alignment. In *International Conference on Artificial Intelligence and Statistics*, pages 2269–2277. PMLR, 2021.

[2] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

[3] R Dennis Cook and Liliana Forzani. On the mean and variance of the generalized inverse of a singular Wishart matrix. 2011.

[4] Christopher Cox and Guangqin Ma. Asymptotic confidence bands for generalized nonlinear regression models. *Biometrics*, pages 142–150, 1995.

[5] Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. *Advances in Neural Information Processing Systems*, 33:5850–5861, 2020.

[6] David V. Hinkley. Jackknifing in unbalanced situations. *Technometrics*, 19(3):285–292, 1977.

[7] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, March 2021.

[8] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[9] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[10] James G MacKinnon and Halbert White. Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of econometrics*, 29(3):305–325, 1985.

[11] Harald Martens and Magni Martens. Modified jack-knife estimation of parameter uncertainty in bilinear modelling by partial least squares regression (plsr). *Food quality and preference*, 11(1-2):5–16, 2000.

[12] Gaétan Monari and Gérard Dreyfus. Withdrawing an example from the training set: An analytic estimation of its effect on a non-linear parameterised model. *Neurocomputing*, 35(1):195–201, November 2000.

[13] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

[14] R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

[15] Daryl Pregibon. Logistic Regression Diagnostics. *The Annals of Statistics*, 9(4):705–724, July 1981. Publisher: Institute of Mathematical Statistics.

[16] Maurice H Quenouille. Notes on bias in estimation. *Biometrika*, 43(3/4):353–360, 1956.

[17] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[18] Jack Sherman and Winifred J Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950.

[19] John W. Tukey. Bias and confidence in not-quite large samples (preliminary report). *Annals of Mathematical Statistics*, 29(2):614, 1958.

[20] Andrew Wang, Elisa Nguyen, Runshi Yang, Juhan Bae, Sheila A. McIlraith, and Roger Grosse. Better Training Data Attribution via Better Inverse Hessian-Vector Products, July 2025. arXiv:2507.14740 [cs].