SECOND-MOMENT LOSS: A NOVEL REGRESSION OBJECTIVE FOR IMPROVED UNCERTAINTIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantification of uncertainty is one of the most promising approaches to establish *safe* machine learning. Despite its importance, it is far from being generally solved, especially for neural networks. One of the most commonly used approaches so far is Monte Carlo dropout, which is computationally cheap and easy to apply in practice. However, it can underestimate the uncertainty. We propose a new objective, referred to as second-moment loss (SML), to address this issue. While the full network is encouraged to model the mean, the dropout networks are explicitly used to optimize the model variance. We analyze the performance of the new objective on various toy and UCI regression datasets. Comparing to the state-of-the-art of deep ensembles, SML leads to comparable prediction accuracies and uncertainty estimates while only requiring a single model. Under distribution shift, we observe moderate improvements. From a safety perspective also the study of worst-case uncertainties is crucial. In this regard we improve considerably. Finally, we show that SML can be successfully applied to SqueezeDet, a modern object detection network. We improve on its uncertainty-related scores while not deteriorating regression quality. As a side result, we introduce an intuitive Wasserstein distance-based uncertainty measure that is non-saturating and thus allows to resolve quality differences between any two uncertainty estimates.

1 INTRODUCTION

Having attracted great attention in both academia and digital economy, deep neural networks (DNNs, Goodfellow et al. (2016)) are about to become vital components of safety-critical applications. Examples are autonomous driving (Pomerleau, 1989; Bojarski et al., 2016) or medical diagnostics (Liu et al., 2014), where prediction errors potentially put humans at risk. These systems require methods that are robust not only under lab conditions (i.i.d. data sampling), but also under continuous domain shifts, think e.g. of adults on e-scooters or growing numbers of mobile health sensors. Besides shifts in the data, the data distribution itself poses further challenges. Critical situations are (fortunately) rare and thus strongly under-represented in datasets. Despite their rareness, these critical situations have a significant impact on the safety of operations. This calls for comprehensive self-assessment capabilities of DNNs and recent uncertainty mechanisms can be seen as a step in that direction.

While a variety of uncertainty approaches has been established, stable quantification of uncertainty is still an open problem. Many recent machine learning applications are e.g. equipped with Monte Carlo (MC) dropout (Gal & Ghahramani, 2016) that offers conceptual simplicity and scalability. However, is tends to underestimate uncertainties thus bearing disadvantages compared to more recent approaches such as deep ensembles (Lakshminarayanan et al., 2017). We propose an alternative uncertainty mechanism. It builds on dropout sub-networks and explicitly optimizes variances (see Fig. 1 for an illustrative example). Technically, this is realized by a simple additive loss term, the *second-moment loss*. To address the above outlined requirements for safety-critical systems, we evaluate our approach systematically w.r.t. continuous data shifts and worst-case performances.

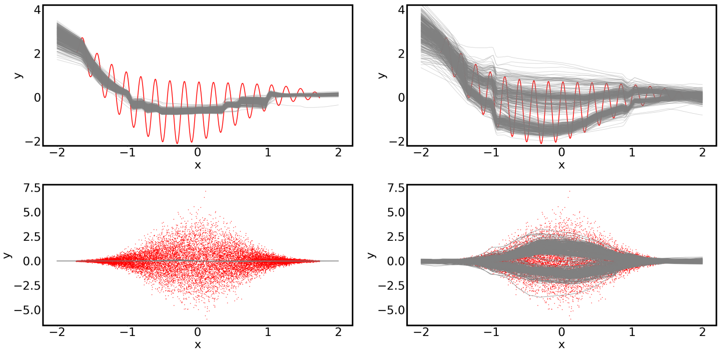

Figure 1: Sampling-based uncertainty mechanisms on toy datasets. The second-moment loss (right) induces uncertainties that capture (de facto) data-inherent uncertainty. This is in contrast to MC dropout (left). 200 sub-networks (grey) and ground truth data (red) are shown.

In detail, our contribution is as follows:

- we introduce a novel regression loss for better calibrated uncertainties applicable to dropout networks,
- we reach state-of-the-art performance in an empirical study and improve on it when considering data shift and worst-case performances, and
- we demonstrate its applicability to real-world applications by example of 2D bounding box regression.

2 Related work

Approaches to estimate predictive uncertainties can be broadly categorized into three groups: Bayesian approximations, ensemble approaches and parametric models.

Monte Carlo dropout (Gal & Ghahramani, 2016) is a prominent representative of the first group. It offers a Bayesian motivation, conceptual simplicity and scalability to application-size neural networks (NNs). This combination distinguishes MC dropout from other Bayesian neural network (BNN) approximations like Blundell et al. (2015) and Ritter et al. (2018). A computationally more efficient version of MC dropout is one-layer or last-layer dropout (see e.g. Kendall & Gal (2017)). Alternatively, analytical moment propagation allows sampling-free MC-dropout inference at the price of additional approximations (e.g. Postels et al. (2019)). Further extensions of MC dropout target tuned performance by learning layer-specific drop rates using Concrete distributions (Gal et al., 2017) and the integration of data-inherent (aleatoric) uncertainty (Kendall & Gal, 2017). Note that dropout training is used—independent from an uncertainty context—for better model generalization (Srivastava et al., 2014).

Ensembles of neural networks, so-called deep ensembles (Lakshminarayanan et al., 2017), pose another popular approach to uncertainty modelling. Comparative studies of uncertainty mechanisms (Snoek et al., 2019; Gustafsson et al., 2020) highlight their advantageous uncertainty quality, making deep ensembles a state-of-the-art method. Fort et al. (2019) argue that deep ensembles capture multi-modality of loss landscapes thus yielding potentially more diverse sets of solutions.

The third group are parametric modelling approaches that extend point estimations by adding a model output that is interpreted as variance or covariance (Nix & Weigend, 1994; Heskes, 1997). Typically, these approaches optimize a (Gaussian) negative log-likelihood (NLL, Nix & Weigend (1994)). A more recent representative of this group is, e.g., Kendall & Gal (2017), for a review see Khosravi et al. (2011). A closely related model class is deep kernel learning. It approaches uncertainty modelling by combining NNs and Gaussian processes (GPs) in various ways, e.g. via an additional layer (Wilson et al., 2016; Iwata & Ghahramani, 2017), by using networks as GP kernels (Garnelo et al., 2018) or by matching NN residuals with a GP (Qiu et al., 2019).

In the context of object detection, various uncertainty approaches can be encountered, e.g. MC dropout in Bhattacharyya et al. (2018) and Miller et al. (2018), or parametric approaches in He et al. (2019). Hall et al. (2020) advocate to account for uncertainty in bounding box detection.

The quality of uncertainties is typically evaluated using negative log-likelihood (Blei et al., 2006; Walker et al., 2016; Gal & Ghahramani, 2016), expected calibration error (ECE) (Naeini et al., 2015; Snoek et al., 2019) and its variants and by considering correlations between uncertainty estimates and model errors, e.g. area under the sparsification error curve (AUSE, Ilg et al. (2018)) for image tasks. Moreover, it is common to study how useful uncertainty estimates are for solving auxiliary tasks like out-of-distribution classification (Lakshminarayanan et al., 2017) or robustness w.r.t. adversarial attacks. An alternative approach is the investigation of qualitative uncertainty behaviors: Kendall & Gal (2017) check if the epistemic uncertainty decreases when increasing the training set or Wirges et al. (2019) studies how the level of uncertainty depends on the distance of the object to a car for some 3D environment regression task.

3 SECOND-MOMENT LOSS

Monte Carlo (MC) dropout was proposed as a computationally cheap approximation of performing Bayesian inference in neural networks (Gal & Ghahramani, 2016). Given a neural network $f_\theta : \mathbb{R}^d \to \mathbb{R}^m$ with parameters θ, MC dropout samples sub-networks $f_{\tilde{\theta}}$ by randomly dropping nodes from the main model f_θ. During MC dropout inference the prediction is given by the mean estimate over the predictions of a given sample of sub-networks, while the uncertainty associated with this prediction can be estimated, e.g. , in terms of the sample variance. During MC dropout training the objective function, e.g. , (in our case) the mean squared error (MSE), is applied to the sub-networks separately. Due to this training procedure, all sub-network predictions are shifted towards the same training targets, which can result in overconfident predictions, i.e. in an underestimation of prediction uncertainty.[1]

Based on this observation, we propose to use the sub-networks $f_{\tilde{\theta}}$ in a different way: they are explicitly *not* encouraged to fit the data mean directly. This is the task of the full network f_θ. The sub-networks $f_{\tilde{\theta}}$ instead model aleatoric uncertainty and prediction residuals if the prediction of the full network f_θ is incorrect. Thus, we deliberately assign different 'jobs' to the main network f_θ on the one hand and its sub-networks on the other hand. Formalizing this idea into an optimization objective yields

$$L = L_{\mathrm{regr}} + L_{\mathrm{sml}} = \frac{1}{M} \sum_{i=1}^{M} \Big[\underbrace{(f_\theta(x_i) - y_i)^2}_{\text{regression loss}} + \beta \underbrace{\left(|f_{\tilde{\theta}}(x_i) - f_\theta(x_i)| - |f_\theta(x_i) - y_i| \right)^2}_{\text{second}-\text{moment loss}} \Big] \ , \quad (1)$$

where the sum runs over a mini-batch of size $M < N$ taken from the set of observed samples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $x_i \in \mathbb{R}^d$ denotes the input, $y_i \in \mathbb{R}^m$ the ground-truth label, and $\beta > 0$ is a hyperparameter that weights both terms. The first term, L_{regr}, is the MSE w.r.t. the full network f_θ. The second term, L_{sml}, seeks to optimize[2] the sub-networks $f_{\tilde{\theta}}$. It aims at finding sub-networks such that the distance $|f_{\tilde{\theta}} - f_\theta|$ matches the aleatoric uncertainty or the prediction residual which is quantified by $|f_\theta(x_i) - y_i|$. As our choice of L_{sml} removes all directional information of the residual, possible (optimal) solutions for the $f_{\tilde{\theta}}$ are not uniquely determined.[3] This leads to a significant increase in the variance of the sub-networks, i.e. the second moment of $f_{\tilde{\theta}}$, compared to standard MC dropout, which is why we name L_{sml} the *second-moment loss* (SML).[4] The standard deviations σ_{total} of the predictions of the sub-networks w.r.t. the prediction of the mean network induced by the SML have two components: the spread σ_{drop} of the sub-networks and an offset $\left|f_\theta - \langle f_{\tilde{\theta}} \rangle\right|$ between the full network and the sub-network mean that our loss might cause, concretely, $\sigma_{\mathrm{total}} = \sigma_{\mathrm{drop}} + |f_\theta - \langle f_{\tilde{\theta}} \rangle|$. While $|f_\theta - \langle f_{\tilde{\theta}} \rangle|$ is reminiscent of residual matching, σ_{drop} seems to be more closely related to

[1] An intuitive explanation is as follows: Let f_θ be a NN with one-dimensional output. For MC dropout with the MSE loss we get $\left\langle (f_{\tilde{\theta}}(x) - y)^2 \right\rangle = (\langle f_{\tilde{\theta}}(x) \rangle - y)^2 + \sigma^2(f_{\tilde{\theta}}(x))$. Therefore, it simultaneously minimizes the squared error between sub-network mean and target and the variance $\sigma^2(f_{\tilde{\theta}}(x)) = \langle f_{\tilde{\theta}}^2(x) \rangle - \langle f_{\tilde{\theta}}(x) \rangle^2$ over the sub-networks.

[2] To avoid unintended optimization of full f_θ in direction of $f_{\tilde{\theta}}$, we only back-propagate through $f_{\tilde{\theta}}$ in L_{sml}.

[3] For a one dimensional example based on aleatoric uncertainty see appendix A.1.

[4] For brevity, we also refer to the entire loss objective L as second-moment loss during evaluation.

modelling uncertainties. We show in appendix A.2 that σ_{drop} accounts on average for more than 80% of σ_{total} in our experiments.

Note that while we investigate the proposed objective in terms of dropout sub-networks in this paper, our arguments as well as the actual approach are generally applicable to other models that allow to formulate sub-networks given some kind of mean model. Besides the regression tasks considered here our approach could be useful for other objectives which use or benefit from an underlying distribution, e.g. uncertainty quantification in classification.

4 Experiments

We begin this section with an illustrative and visualizable toy dataset and continue with benchmarks on various UCI datasets (Dua & Graff, 2017) in section 4.2. To conclude in 4.3, the second-moment loss is applied to a more complex task: object detection in the form of a 2D bounding box regression using the compact SqueezeDet architecture (Wu et al., 2017).

For the first two parts we use an identical set-up of 2 hidden layers with 50 neurons each and ReLu activations. As benchmark methods we consider: MC dropout (abbreviated as **MC**), last-layer MC dropout (**MC-LL**), parametric uncertainty (**PU**), deep ensembles with (**DE-PU**) and without (**DE**) explicit PU. While the toy model has a stronger focus on visual inspection, the UCI evaluation relies on a variety of measures: root-mean-square error (**RMSE**), negative log-likelihood (**NLL**), expected calibration error (**ECE**), and a novel usage of the Wasserstein distance (**WS**). Further details on the network, training procedure, the implementation of methods and measures can be found in appendix B.1. The same holds for the more elaborate SqueezeDet architecture and modifications to it, see B.5.

4.1 Toy datasets

To illustrate qualitative behaviors of the different uncertainty techniques, we consider two $\mathbb{R} \to \mathbb{R}$ toy datasets. This benchmark puts a special focus on the handling of data-inherent uncertainty. The first dataset is Gaussian white noise with an x-dependent (non-linear) amplitude, see first row of Fig. 2. The second dataset is a polynomial overlayed with a high-frequency, amplitude-modulated sine, see fourth row of Fig. 2. The explicit equations for the toy datasets used here can be found in appendix B.2. While the uncertainty in the first dataset ('toy-noise') is clearly visible, it is less obvious for the fully deterministic second dataset ('toy-hf'). There is an effective uncertainty though, as the shallow networks employed are empirically not able to fit (all) fluctuations of 'toy-hf' (see fifth row of Fig. 2). One might (rightfully) argue that this is a sign of insufficient model capacity. But, in more realistic, e.g., higher dimensional and sparser datasets the distinction between true noise and complex information becomes exceedingly difficult to make. As the Nyquist-Shannon sampling theorem states, with limited data deterministic fluctuations above a cut-off frequency can no longer be resolved (Landau, 1967). They therefore become virtually indistinguishable from random noise.

The mean estimates of all uncertainty methods (second and fifth row in Fig. 2) look alike on both datasets. They approximate the noise mean and the polynomial, respectively. In the latter case, all methods rudimentarily fit some individual fluctuations. The variance estimation (third and sixth row in Fig. 2) in contrast reveals significant differences between the methods: While PU, PU-DE, and the network trained with SML are capable of capturing aleatoric uncertainty, MC dropout variants and non-parametric ensembles are not. This behavior of MC dropout is expectable as it was introduced to account for model uncertainty not data-inherent uncertainty. The non-parametric ensemble is effectively optimized in a similar fashion. In contrast, NLL-optimized PU networks have a home-turf advantage on these datasets since the parametric variance is explicitly optimized to account for the present aleatoric uncertainty. The SML provides comparably good uncertainty estimates. They are evoked by the L_{sml}-term that incentivizes sub-networks $f_{\tilde{\theta}}$ to keep an adequate distance from f_θ. While the outcomes of both PU (PU-DE) and SML-trained network look similar, the mechanics of the two approaches are fundamentally different. We investigate the drivers behind the adjustments of the sub-networks in appendices A and A.3. Accompanying quantitative evaluations can be found in appendix B.2.

In the following, we substantiate the corroborative results of the SML on toy data by an empirical study on UCI datasets and an application to a modern object detection network.

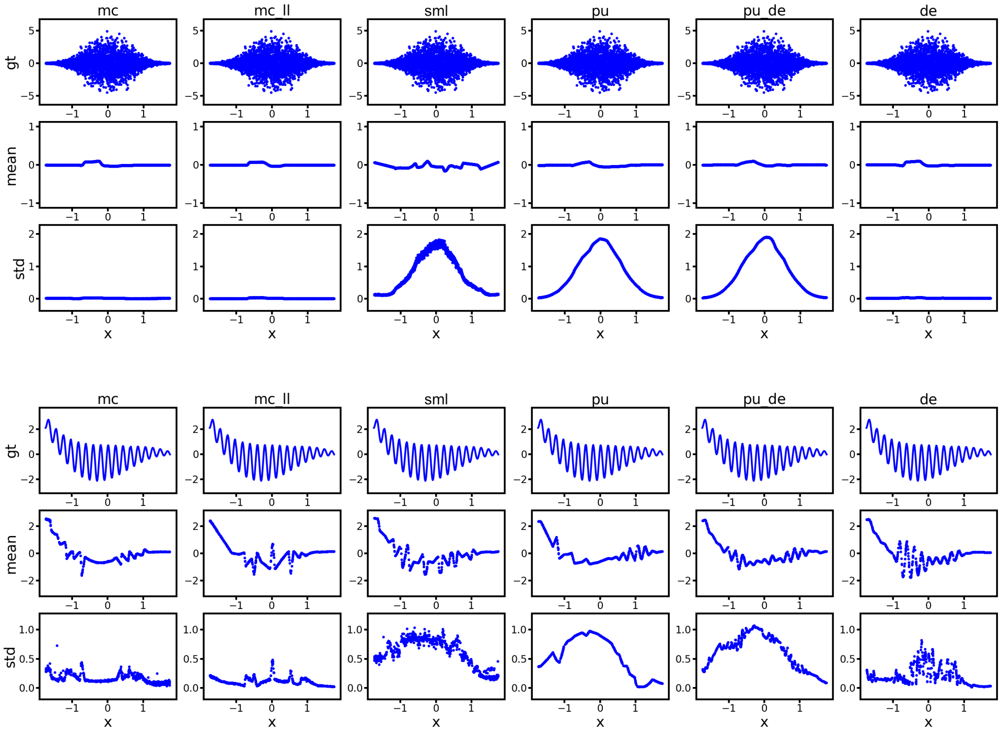

Figure 2: Comparison of uncertainty approaches (columns) on two 1D toy functions: a noisy one (top) and a high-frequency one (bottom). Test data ground truth (respective first row) is shown with mean estimates (resp. second row) and standard deviations (resp. third row).

4.2 UCI REGRESSION DATASETS

Next, we study UCI regression datasets, extending the dataset selection in Gal & Ghahramani (2016) by adding three further datasets: ‘diabetes’, ‘california’, and ‘superconduct’. Apart from train- and test-data results, we study regression performance and uncertainty quality *under data shift*. Such distributional changes and uncertainty quantification are closely linked since the latter ones are rudimentary “self-assessment” mechanisms that help to judge model reliability. These judgements gain importance for model inputs that are *structurally different* from train data. Appendix B.3 elaborates on different ways of splitting the data, namely *pca-based* splits in input space (using the first principal component) and *label-based* splits. More general information on training and dataset-dependent modifications to the experimental setup are relegated to the technical appendix B.1. For brevity of exposition, we limit our discussion here largely to the ECE and the worst-case uncertainty performance. An evaluation of the remaining measures, including the Wasserstein measure, is given in appendix B.3. All presented results are 5- or 10-fold cross validated.

Fig. 3 provides ECEs for 13 UCI datasets that are sorted by dataset size on the x-axis. The top panel shows train (green) and test set (blue) ECEs, the bottom panel test set ECEs under two pca-based data shifts (yellow-green, orange) and two label-based data splits (red, light red), respectively. Uncertainty methods are encoded via plot marker, e.g. PU-DE as ‘star’ and SML-trained networks (‘ours’) as ‘square’. We summarize these dataset-specific results on the right hand side of Fig. 3 (light grey background). The columns ‘mean’ and ‘median’ of this summary show that on training sets, ECEs are smallest for PU, followed by PU-DE and the SML network. On test data, however, PU, PU-DE and the SML network share the first place. Looking at the stability w.r.t. data shift, i.e. extra- and interpolation based on label-split or pca-split, PU loses in performance while PU-DE and SML reach the smallest calibration errors in three out of four cases.

For NLL and Wasserstein measure, PU-DE and the SML-trained network reach comparably small average values with advantages for SML-trained network under data shift, see Fig. 9 and Fig. 10

in appendix B.3 for detailed evaluations. In contrast to uncertainty quality, regression performances are almost identical for all uncertainty methods (see Fig. 8 and Table 5 in appendix B.3).

Summarizing these evaluations on UCI datasets, we find SML to be as strong as the state-of-the-art method of PU-DEs while using only a single network compared to an ensemble of 5 networks. We moreover observe advantages for SML under PCA- and label-based data shifts. Three datasets lead to overestimated uncertainties for the SML, see discussion in appendix B.3. A visual tool to further inspect uncertainty quality are residual-uncertainty scatter plots as shown in appendix B.4. For a reflection on NLL and comparisons of the different uncertainty measures on UCI data see again appendix B.3.

From a safety perspective the study of worst-case uncertainties is crucial. A better understanding of these least appropriate uncertainties might allow to determine lower bounds on operation quality of safety-critical systems. We restrict our analysis to uncertainty estimates that *under-estimate* prediction residuals, i.e. $|r_i| \gg 1$. These cases might be more harmful than overly large uncertainties, $|r_i| \ll 1$, that likely trigger a conservative system behavior. We quantify worst-case uncertainty performance as follows: for a given (test) dataset, the absolute normalized residuals $\{|r_i|\}_i$ are calculated. We determine the 99% quantile $q_{0.99}$ of this set and calculate the mean value over all $|r_i| > q_{0.99}$, the so-called expected tail loss at quantile 99% ($\mathbf{ETL_{0.99}}$) (Rockafellar & Uryasev, 2002). The ETL$_{0.99}$ measures the average performance of the worst performing 1%.

For both toy datasets and 12 UCI datasets, the test data ETL$_{0.99}$'s of all trained network are calculated, yielding a total of 105 ETL$_{0.99}$ values per uncertainty method. Table 1 reports the mean value and the maximal value of these ETL$_{0.99}$'s for PU-DE and SML-trained networks as these two methods show the strongest performances throughout this work. While none of these methods gets close to the ideal ETL$_{0.99}$'s of a $\mathcal{N}(0, 1)$, SML-trained networks exhibit significantly less pronounced tails and therefore higher stability compared to PU-DE. This holds true over all considered test sets. Deviations from standard normal grow from i.i.d. test over PCA-based train-test split to label-based train-test split. We attribute the lower stability of PU-DE to the nature of the PU networks composing the ensemble. The inherent instability of parametric uncertainty estimation (see Table 5 in appendix B.3) is largely suppressed by ensembling. Considering the tail of the $|r_i|$-distribution however reveals that regularization of PU by ensembling works not in every single case. It is unlikely that larger ensemble are able to fully cure this instability issue. SML-trained networks in contrast encode uncertainty into the structure of the entire network thus yielding preferable stability compared to parametric approaches.

Table 1: Worst-case uncertainty quality for different uncertainty methods: SML-induced uncertainties (ours) and PU-DE are compared to the ideal Gaussian case for i.i.d. and non-i.i.d. data splits. Worst-case uncertainty quality is quantified by the expected tail loss at the 99% quantile (ETL$_{0.99}$). Each mean and max value is taken over the ETLs of 105 models trained on 14 different datasets.

measure	data split	$\mathcal{N}(0, 1)$	Ours	PU-DE
mean ETL$_{0.99}$	i.i.d.	2.89	3.80	5.03
max ETL$_{0.99}$	i.i.d.	3.01	9.71	19.69
mean ETL$_{0.99}$	pca	2.89	4.62	6.71
max ETL$_{0.99}$	pca	3.01	13.0	39.34
mean ETL$_{0.99}$	label	2.89	5.18	38.65
max ETL$_{0.99}$	label	3.01	35.96	799.78

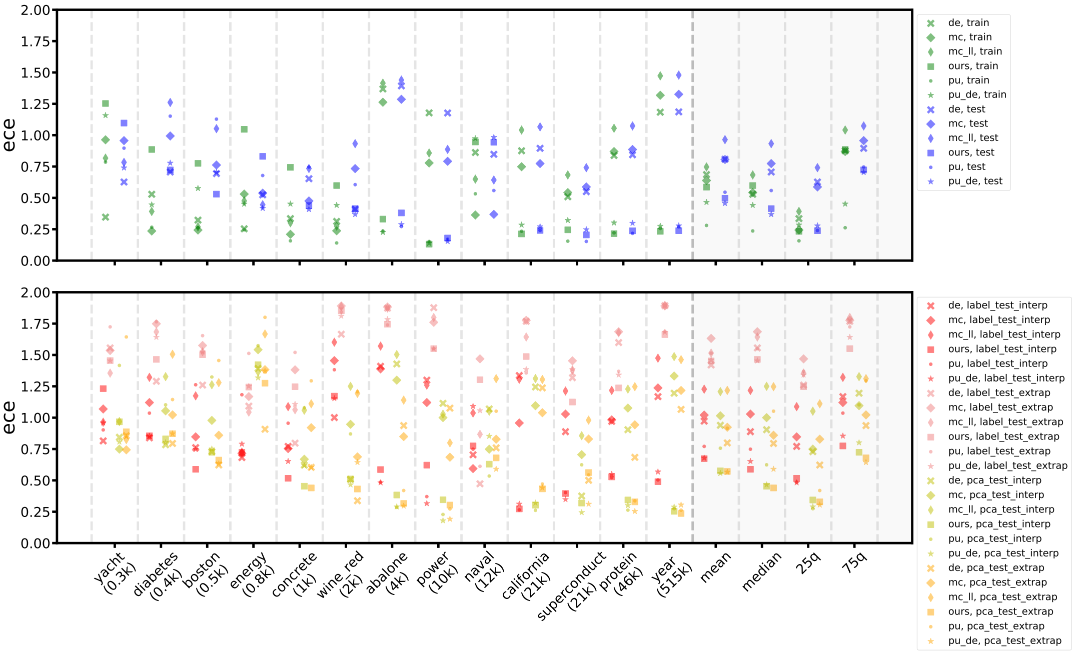

Figure 3: Expected calibration errors (ECEs) for 13 UCI regression datasets under i.i.d. conditions (top) and under data shift (bottom). Uncertainty methods are encoded via plot marker, data splits via color. Each plot point corresponds to a cross-validated trained network. Summarizing statistics (rhs) are indicated by a light grey background.

4.3 Application to object regression

After studying toy and UCI datasets, we turn towards the challenging real-world task of object detection, namely the SqueezeDet model (Wu et al., 2017), a fully convolutional neural network. It is trained and evaluated on KITTI (Geiger et al., 2012). For details on the SqueezeDet architecture and the KITTI data split, see B.5. We compare standard SqueezeDet with SML-SqueezeDet that uses the second-moment loss instead of the original MSE regression loss (see appendix B.5 for more details). In both settings the model is trained for $150{,}000$ mini-batches of size 20, i.e. for 815 epochs. After training, we keep dropout active and compute 50 forward passes for each test image. For standard SqueezeDet, all forward passes are individually matched with ground truth. We exclude predictions from the evaluation if their IoU with ground truth is ≤ 0.1. While standard SqueezeDet (with activated dropout at inference) uses the mean of the dropout samples for prediction, SML-SqueezeDet uses the full network instead (see section 3). These predictions and their corresponding dropout samples are matched based on the respective anchor. The dropout samples are summarized by their means and variances.

To assess model performance, we report the mean intersection over union (mIoU) and RMSE (in pixel space) between predicted bounding boxes and matched ground truths. The quality of the uncertainty estimates is measured by (coordinate-wise) NLL, ECE and Wasserstein distance. Table 2 shows a summary of our results on train and test data. The results for NLL, ECE and WS have been averaged across the 4 regression coordinates. SqueezeDet and SML-SqueezeDet show comparable regression results, with slight advantages for SML-SqueezeDet on test data. Considering uncertainties quality, we find substantial advantages for SML-SqueezeDet across all evaluation measures. These findings resemble those on the UCI regression datasets and indicate that the second-moment loss works well on a modern application-scale network.

5 Conclusion

We approach dropout-based uncertainty quantification from a new direction: sub-networks are explicitly not encouraged to model the data mean, they capture data-inherent uncertainties and po-

Table 2: Regression performance and uncertainty quality of SqueezeDet-type networks on KITTI test data. SML-trained SqueezeDet (ours) is compared with the default SqueezeDet that uses one-layer dropout to estimate uncertainties. The measures of NLL, ECE and WS are aggregated along their respective four dimensions, for details see appendix B.5 and table 6 therein.

measure	SqueezeDet	SML-SqueezeDet	SqueezeDet	SML-SqueezeDet
	train		test	
mIoU (↑)	0.816	0.812	0.738	0.744
RMSE (↓)	6.418	6.862	18.225	17.492
NLL (↓)	20.746	3.916	98.807	17.875
ECE (↓)	0.996	0.554	1.198	0.834
WS (↓)	2.487	0.874	4.587	1.734

tential fitting residuals of the full network instead. Technically, this is realized by an additional loss term that accompanies the standard regression objective: the *second-moment loss*. Our loss enables stable training. Training complexity and runtime behavior at inference are comparable to MC dropout. Task performances and uncertainty qualities of these models are on par with (parametric) deep ensembles, the widely used state-of-the-art for uncertainty quantification. However, unlike deep ensembles, we use single networks. In practice, this might allow to reduce training effort significantly compared to deep ensembles, especially for application-scale networks. Moreover, a single network requires only a fraction of the storage of a deep ensemble, making models with competitive uncertainties more accessible for mobile or embedded applications.

An extensive study of uncertainties under data shift revealed advantages of SML-trained models compared to deep ensembles: while both methods *on average* provide comparable results, we find a higher stability across a variety of datasets and data shifts. With respect to worst-case uncertainties SML-trained networks are by a large margin better than deep ensembles. A quite relevant finding for safety-critical applications like automated driving or medical diagnosis where (even rarely occurring) inadequate uncertainty estimates might lead to injuries and damage. Technically, we attribute this gain in stability to our sub-network-based approach: like MC dropout, we integrate uncertainty estimates into the very structure of the network, rendering it more robust towards unseen inputs than a parameter estimate.

Moreover, the second-moment loss can serve as a general drop-in replacement for MC dropout on regression tasks. For already trained MC dropout models, post-training with the second-moment loss might suffice to improve on uncertainty quality. As an outlook, our first such post-training experiments on UCI datasets are encouraging. Another interesting variant is the combination of SML with last-layer dropout as it enables sampling-free inference (Postels et al., 2019). Preliminary experiments on UCI datasets show clearly improved uncertainties qualities compared to standard MC-LL. A potentially interesting avenue for near-real time applications.

The simple additive structure of the second-moment loss makes it applicable to a variety of optimization objectives. For classification, we might be able to construct a non-parametric counterpart to prior networks (Malinin & Gales, 2018). Taking a step back, we demonstrated an easily feasible approach to influence and train sub-network distributions. This could be a promising avenue, for distribution matching but also for theoretical investigations.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4194–4202, 2018.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

David M Blei, Michael I Jordan, et al. Variational inference for Dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Andrew YK Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. 'in-between'uncertainty in Bayesian neural networks. *arXiv preprint arXiv:1906.11537*, 2019.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pp. 1050–1059, 2016.

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in Neural Information Processing Systems*, pp. 3581–3590, 2017.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. volume 15 of *Proceedings of Machine Learning Research*, pp. 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. JMLR Workshop and Conference Proceedings.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable Bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 318–319, 2020.

David Hall, Feras Dayoub, John Skinner, Haoyang Zhang, Dimity Miller, Peter Corke, Gustavo Carneiro, Anelia Angelova, and Niko Sünderhauf. Probabilistic object detection: Definition and evaluation. In *The IEEE Winter Conference on Applications of Computer Vision*, pp. 1031–1040, 2020.

Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2888–2897, 2019.

Tom Heskes. Practical confidence and prediction intervals. In *Advances in Neural Information Processing Systems*, pp. 176–182, 1997.

Eddy Ilg, Ozgun Cicek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 652–667, 2018.

Tomoharu Iwata and Zoubin Ghahramani. Improving output uncertainty estimation and generalization in deep learning via neural network Gaussian processes. *arXiv preprint arXiv:1707.05922*, 2017.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pp. 5574–5584, 2017.

Abbas Khosravi, Saeid Nahavandi, Doug Creighton, and Amir F Atiya. Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Transactions on Neural Networks*, 22(9):1341–1356, 2011.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *35th International Conference on Machine Learning, ICML 2018*, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

HJ Landau. Sampling, data transmission, and the Nyquist rate. *Proceedings of the IEEE*, 55(10): 1701–1706, 1967.

Siqi Liu, Sidong Liu, Weidong Cai, Sonia Pujol, Ron Kikinis, and Dagan Feng. Early diagnosis of alzheimer's disease with deep learning. In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pp. 1015–1018. IEEE, 2014.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, pp. 7047–7058, 2018.

Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7. IEEE, 2018.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2015, pp. 2901, 2015.

David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of IEEE International Conference on Neural Networks 1994*, volume 1, pp. 55–60. IEEE, 1994.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pp. 305–313, 1989.

Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2931–2940, 2019.

Xin Qiu, Elliot Meyerson, and Risto Miikkulainen. Quantifying point-prediction uncertainty in neural networks via residual estimation with an i/o kernel. *arXiv preprint arXiv:1906.00588*, 2019.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable Laplace approximation for neural networks. *ICLR*, 2018.

R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, 2002.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pp. 59, USA, 1998.

Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pp. 13969–13980, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Michael A Stephens. EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737, 1974.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pp. 835–851. Springer, 2016.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pp. 370–378, 2016.

Sascha Wirges, Marcel Reith-Braun, Martin Lauer, and Christoph Stiller. Capturing object detection uncertainty in multi-layer grid maps. *arXiv preprint arXiv:1901.11284*, 2019.

Bichen Wu, Forrest Iandola, Peter H Jin, and Kurt Keutzer. SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 129–137, 2017.

Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Data-driven 3d voxel patterns for object category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1903–1911, 2015.