Seminar Thesis

Uncertainty Estimation in OOD Settings: Comparing Neural Networks and Bayesian Last-Layer Approaches

Department of Statistics
Ludwig-Maximilians-Universität München

Bakir Chaban

Munich, July 3rd, 2025

Submitted in partial fulfillment of the requirements for the degree of B. Sc.
Supervised by Prof. Dr. Ludwig Bothmann

Abstract

This thesis aims to introduce partially stochastic neural networks with a Bayesian last-layer as a method to capture uncertainty in predictions. Uncertainty estimation is essential for improving the reliability of machine learning models, particularly when dealing with out-of-distribution (OOD) data. The objective is to compare a deterministic neural network with a Bayesian last-layer network by conducting an experiment on synthetic and real-world data with OOD samples added. Key metrics including prediction interval coverage, predicted standard deviation, mutual information and Shannon entropy were utilized.

The results indicate that incorporating Bayesian inference in the last layer of a neural network improves uncertainty estimation on OOD data. In contrast, the deterministic network, fails to capture uncertainty and remains overconfident on OOD data. The Bayesian last-layer network demonstrates improved Shannon entropy separation and a lower predicted mean probability.

These findings support the effectiveness of Bayesian last-layer networks in the context of uncertainty estimation. Nonetheless, the results also reveal limitations, indicating the need for further refinement.

Remark

This seminar thesis was created with partial assistance of artificial intelligence tools. More specifically, ChatGPT has been used to assist in coding and debugging, clarifying questions on this topic as well as gathering additional information. Additionally, Grammarly has been used to check for grammatical errors.

Contents

1 Introduction

Whether used in computer vision, object detection, or the medical field, deploying neural networks to obtain estimates and generate predictions is a standard approach in the machine learning and data science community (Caruana et al., 2015, Girshick, 2015). Nonetheless, being confronted with out-of-distribution (OOD) and noisy data is a common co-occurrence in the majority of real-world applications.

Solely relying on point estimates and excluding uncertainty introduced by the data or the model, however, could result in misinterpretation of the predictions. Receiving overconfident or wrong predictions, especially when used in the medical field, e.g., when diagnosing a disease could lead to deteriorating events (Caruana et al., 2015).
Additionally, it becomes an issue when confronted with test sets that do not lie in the feature space of the trained model, thus, affecting prediction capabilities of convolutional neural networks (Korotin et al., 2020).

A viable method to capture uncertainty would be to introduce Bayesian Neural Networks, more specifically Bayesian last-layer neural networks.
By using these approaches, one provides a way to include uncertainty in the estimations and, thus, enable the end user to either rework the model parameters or prior assumptions. Moreover, the uncertainty of the prediction introduced by OOD data could now be successfully captured, resulting in less confident predictions (Kristiadi et al., 2020).

This paper introduces the theory behind partial Bayesian neural networks with a Bayesian last-layer. In addition to that, we conduct a comparison between a complete deterministic neural network and a Bayesian last-layer network. The models are evaluated on a regression task using a synthetic dataset and a classification task using a dataset containing blood samples indicative of diabetes, both tasks involving OOD data sampled from different distributions.
The aim is to highlight the advantage of a Bayesian last-layer network when used on OOD data, compared to a standard neural network.

2 Related Work

Overconfidence in ReLU Networks
Hein et al. (2019) focused on addressing the problem of overconfident predictions on faraway data in the context of ReLU networks. In their theoretical findings, the overconfidence arises from the fact that there exist infinitely many inputs that result in high softmax outputs and therefore overconfidence. Post hoc methods such as temperature scaling failed to yield substantial improvement.
Nonetheless, Hein et al. (2019) state that the only viable solution to improve highly confident predictions is to modify the networks' architecture. They, however, do not provide a specific alternative, leaving open the question of which architecture mitigates this problem. This is where stochastic neural networks or Bayesian neural networks come into focus.

Bayesian Last-Layer
Kristiadi et al. (2020) highlight that by being Bayesian in the last layer of a neural network, that is, by assuming a prior (Gaussian) distribution on the weights instead of a single value, one can reduce the confidence in faraway data. Moreover, they have analytically shown that standard neural networks with MAP estimates achieve overly high asymptotic confidence for a scaled input, as opposed to a Bayesian last-layer (BLL) network, which depicts a decrease in asymptotic confidence. It is to mention that by only capturing uncertainty in the last layer, one ignores uncertainty introduced by the previous layers. Proposing methods such as spectral normalized Gaussian processes (SNGP) could address this issue. Kristiadi et al. (2020) also state that there is no advantage in utilizing a fully Bayesian Neural Network.

Full versus partial stochasticity
Sharma et al. (2023) further investigate this finding. A benchmark on predictive performance of fully stochastic networks against partially stochastic networks across multiple test cases were conducted, indicating no necessity for a complete stochastic network. They report that partially stochastic networks can match and outperform fully stochastic networks while requiring less memory and less training time.

Positioning of this thesis
Based on these studies, this thesis evaluates a deterministic neural network and a BLL network under OOD conditions in both regression and classification. The objective is to empirically validate the theoretical property of a BLL network and assess the effectiveness in improving uncertainty estimation compared to a deterministic neural network.

3 Methodology[1]

3.1 Neural Networks

A fully connected feed-forward neural network is comprised of neurons. Each neuron is a linear combination of the output from the previous layer, the parameters $\theta = (W,b)$ which correspond to the weight matrix and bias vector, the input x and the activation function ϕ.
The simplest version being a multilayer perceptron (MLP) that receives an input vector of dimension d and outputs a vector of dimension k. The MLP consists of L nested linear functions with $L+1$ nonlinear activation functions as shown in (1).

$$f : \mathbb{R}^d \to \mathbb{R}^k, \qquad f(x;\theta) = \phi\Big(W_L\,\phi\Big(W_{L-1}\,\phi\big(\cdots\phi(W_1x + b_1)\cdots\big)\Big)\Big) \tag{1}$$

Without nonlinear activations, the network remains a nested linear model, and no advantage in capturing complexity is gained. However, with the addition of nonlinearity, neural networks can be applied to more demanding tasks. Commonly used activation functions for an input x are listed in Table A.1.

In order to achieve estimates, performing forward feeds through the network, where parameter values are set and tested, is necessary. Maximum likelihood estimation (MLE) is a standard method to determine the parameters.
Bridging the gap to the following section about Bayesian deep learning (BDL), it is shown in Krause and Hübotter (2025), that by adding a Gaussian prior to the MLE and therefore adding regularization, one obtains a maximum-a-posteriori (MAP) estimate. It is described as achieving a single estimate from the posterior.

In addition, the MLE equates to a MAP estimate with a uniform prior, since scaling the maximum of the posterior by a constant scalar, leaves the maximization argument of the MAP unchanged. Once the loss function has been defined and the first forward feed in the network has occurred, backpropagation is deployed to optimize the parameters using popular methods such as stochastic gradient descent (SGD) or ADAM.
Nonetheless, since a MAP estimate remains a point estimate from the posterior, we are still limited in explaining uncertainty, which BDL addresses.

[1]The formulas and definitions are derived from the following sources (Murphy, 2022, 2023, Krause and Hübotter, 2025)

3.2 Bayesian Deep Learning

In BDL the goal is to find the posterior distribution given a training dataset D. By placing a prior over the parameters of the network, they are therefore treated as random variables. Due to this method, we are then able to capture uncertainty in the predictions. The predictive posterior distribution, given (2), can be computed by using Bayesian model averaging, which is explained in the following subsection.

$$p(y|x, D) = \int p(y|x, \theta)p(\theta|D)d\theta \tag{2}$$

In (2), $p(y|x, \theta)$ corresponds to the likelihood of observing the output y given input x and model parameters θ, whereas $p(\theta|D)$ is the posterior which is proportional to the product of the likelihood times the prior $\propto p(\theta)p(D|\theta)$.

Usually a Gaussian prior and a Gaussian likelihood are used as the standard distribution family. As aforementioned, approximation methods are necessary, since the true posterior is intractable. For this there exist different approximation methods such as Laplace approximation, ensemble methods, Monte Carlo (MC) dropout and last-layer methods. We deploy the last-layer method, which is explained in the following subsection.

3.3 Approximation Methods

Bayesian Last-Layer

A common approach to approximate the posterior in BDL is to be Bayesian on the last layer of the network. It corresponds to performing MAP point estimates on $L-1$ weights and placing a prior on the weights in the final layer of the network, resulting in a Bayesian generalized linear model (GLM) (Murphy, 2023). As mentioned in (Riquelme et al., 2018), it is also titled a neural-linear approximation.

$$z = w_L^T \phi(x, \theta) \tag{3}$$

In (3), $\phi(x, \theta)$ corresponds to the deterministic logits obtained from the $L-1$ previous layers and w_L^T as the weights from the last layer.

As stated in Sharma et al. (2023), the training of the BLL network is performed by minimizing the negative log-likelihood (NLL) of the deterministic weights in the previous $L-1$ layers. Once the weights have been optimized, they are then frozen so that only the parameters in the last layer are optimized.
The posterior over the last-layers' weights can then be approximated using Variational Inference (VI), specifically Bayes by Backprop, which will be explained in the following subsection. This training strategy is considered a two stage training method as mentioned in Sharma et al. (2023). Due to the simple implementation, this training procedure was selected in the experiment in section 4.

While a BLL provides an approach to introducing uncertainty estimation in neural networks, there exists a trade-off of deploying this method. On the one hand, last-layer Bayesian inference requires less memory than a fully Bayesian neural network and, therefore, allows for shorter training time (Sharma et al., 2023).

In addition, it can be easily integrated in existing networks, since only the last layer would be replaced, highlighting its versatility.
On the other hand, stochasticity only in the last layer is restricted in capturing all of the networks' uncertainty as explained in Murphy (2023), potentially leading to over- or underestimation of the total uncertainty.
Furthermore, the choice of the prior distribution as well as suitable approximation methods for the posterior affect the performance of the BLL network. Additionally, it is important to clarify that by approximating the posterior over the last-layers' weights, one does not approximate the full network posterior (Sharma et al., 2023).

Variational Inference
Variational methods approximate the posterior by choosing a distribution family for the prior and minimizing the Kullback-Leibler (KL) divergence between the true posterior $p(\theta|D)$ and distributions q; see (4). In the context of Bayesian neural networks the method is referred to as Bayes by Backprop, where independent Gaussian distributions are used (Krause and Hübotter, 2025). Since the variational distribution assumes independence among parameters, the optimization becomes more efficient and requires fewer parameters to learn. This corresponds to mean-field variational inference (MFVI) (Murphy, 2023). Despite its limitations in uncertainty estimation, as demonstrated by Foong et al. (2019), this method is simpler to implement than a Laplace approximation and was therefore chosen in our case.

$$\arg\min_{q \in Q} \mathrm{KL}(q||p(\theta|D)) \tag{4}$$

Named Evidence Lower Bound (ELBO), maximizing the ELBO minimizes the KL divergence; see (5). The KL term regularizes by pulling the variational posterior towards the prior. If the influence is too dominant, the posterior can collapse and one would lose information.

$$\mathbb{E}_q\big[\log p(y|x,\theta)\big] - \mathrm{KL}(q|\, p(\theta|D)) \tag{5}$$

When applying VI, Krause and Hübotter (2025) provide proof for the equivalence on minimizing the cross entropy (CE) in a classification and the mean squared error (MSE) in a regression case, to the MLE, which can be substituted with the log-likelihood in the ELBO. This is of importance when applying the aforementioned methods. For one to be able to perform backpropagation to update our parameters, or in this case distributions over continuous variables, one needs to utilize the reparameterization trick as explained in (Murphy, 2023).

$$p(y|x,D) \approx \frac{1}{m}\sum_{i=1}^{m} p(y|x,\theta^i) \tag{6}$$

Having approximated the true posterior with the variational posterior, see (6), one can now sample m i.i.d $\theta^{(i)}$ using the Monte Carlo (MC) sampling method. Utilizing the samples, one can then estimate the mean and variance of the predictions (Krause and Hübotter, 2025).

The Variance using the law of total variance can be dismantled into the expected variance and the variance of the expected prediction. The first part corresponds to the aleatoric and the second part to the epistemic uncertainty (Krause and Hübotter, 2025). The topic of uncertainty will be briefly covered by the following subsection.

3.4 Uncertainty

Generally, sources of uncertainty are divided into two parts. Firstly, aleatoric uncertainty, explaining randomness originating from statistical processes. Usually, random noise is a common source of the aleatoric type, which is assumed to be irreducible.
Secondly, uncertainty of the epistemic nature originates from restricted knowledge of either the appropriate model parameters or lack of available data. Hence, it can be reduced by including more information (Hüllermeier and Waegeman, 2021).

Lastly, being able to quantify uncertainty is particularly valuable, especially since it is an active research field and integrated into different areas such as medicine (Seoni et al., 2023). However, it needs to be mentioned that the available metrics in Table A.2, provides an additivity given that the Shannon entropy is defined as adding the expected entropy and mutual information. This assumption is still being examined in current research (Wimmer et al., 2023).

Creating the bridge to distribution shifts, Hein et al. (2019) shows that overconfidence on unseen or far away data is apparent in some NN architectures, which is why we transition into the last section covering distribution shifts and OOD data.

3.5 Distribution Shifts

Distribution shifts potentially occur, when given test data is not from the same distribution as the training data. These shifts appear in different ways, i.e., concept shifts or label shift. Manually adding random noise would also be considered a distribution shift, which directly affects prediction performance as shown in Murphy (2023). OOD data also falls into the same category, negatively impacting predictions (Kristiadi et al., 2020).

To conclude, we now combine all the previously explained topics and attempt to cover those in the following conducted experiment, comparing neural network architectures on their ability to quantify uncertainty when tested with OOD data.

4 Experiment

The main goal of the experiment is to assess how two different neural network architectures perform with respect to predictive uncertainty on OOD data. We compare a BLL network with $L-1$ deterministic layers followed by a Bayesian last-layer against a fully deterministic neural network with MAP estimates. Two tasks are considered, where the first task is a regression using a one-dimensional synthetic dataset and the second task a multiclass classification using blood sample data, both with manually introduced OOD data.

4.1 Simulated Regression

We initialized our deterministic neural network with an input layer of size one, followed by two hidden layers of sizes 32 and one hidden layer of size 16 and an output layer of size one. Equal amounts of hidden layers and dimensions were used to define the BLL network. The dataset was generated by sampling 900 instances, where 800 were used for training and 100 for testing; see (7).

$$\begin{aligned} x_i &\overset{\text{iid}}{\sim} \text{Uniform}\big([-4,\, 4]\big), \\ \varepsilon_i &\overset{\text{iid}}{\sim} \mathcal{N}\big(0,\; 0.4^2\big), \\ y_i &:= \sin(x) + \varepsilon_i. \end{aligned} \tag{7}$$

A total of 100 OOD instances were generated by sampling from a normal distribution and then appended to the test set; see (8).

$$\begin{aligned} x_i^{\text{OOD,L}} &\sim \mathcal{N}\big(-6, 1^2\big) \;\Big|\; x_i^{\text{OOD,L}} < -4, \\ x_i^{\text{OOD,R}} &\sim \mathcal{N}\big(\;6, 1^2\big) \;\Big|\; x_i^{\text{OOD,R}} > 4. \end{aligned} \tag{8}$$

Both the deterministic network and shared base network as well as the BLL were trained for 100 epochs with a batch size of 32. The ADAM optimizer was used with a learning rate of 0.001, in order to allow for a faster convergence and enough exploitation, avoiding elongating the optimizing duration. Due to restricted computational resources, we limited the number of MC samples drawn from the variational posterior of the BLL to 100. The MSE was used as the likelihood term in the ELBO. The prior parameters for the BLL were manually tuned based on validation performance, resulting in prior variances of -0.5 for both weights and biases, and a prior standard deviation (std) of 0.5, used in the KL-divergence term of the ELBO.

Table 1: Regression performance of the deterministic neural network and BLL network on ID and OOD data using different metrics.

Model	Region	RMSE	NLL	PI cov	Avg σ
MAP	ID	1.03	3.32	0.53	0.40
MAP	OOD	1.86	10.80	0.25	0.40
BLL	ID	**1.11**	**1.57**	**0.87**	0.84
BLL	OOD	2.63	**2.43**	**0.90**	**1.79**

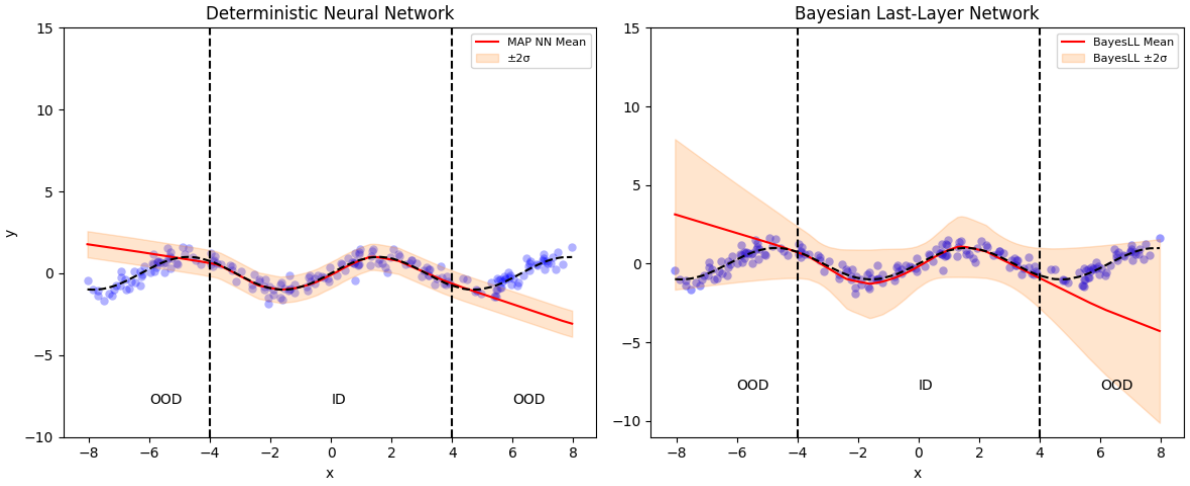

Figure 1: Predictive mean and std bands for the deterministic and BLL network on a regression task

Applying both models on our modified test set yielded the following results. As shown in Table 1, the BLL network performed similarly well with respect to the RMSE on ID but poorer on OOD data. Furthermore, it achieved a lower negative log likelihood (NLL) and improved prediction interval (PI) coverage both on ID (0.87) and OOD (0.90) data. Finally, the BLL network exhibited an increase in average predictive std from 0.87 on ID to 1.78 on OOD data.
In contrast, the deterministic network displayed a drop in PI coverage from 0.52 on ID to 0.24 on OOD data. In addition, the network produced a constant std of 0.4. Figure 1 provides a visual comparison, where the confidence band of the deterministic network is constant, as opposed to a widening confidence band observed for the BLL network on OOD data.

4.2 Classification

In this task we performed a multiclass classification with 3 target categories [N: Non-diabetic, P: Pre-diabetic, Y: Diabetic], using a dataset containing blood markers from anonymous patients. The dataset consisted of 1000 observations and 11 features, including variables such as Gender, Cholesterol, HDL and LDL. The numeric columns were standardized to maintain a mutual scale.

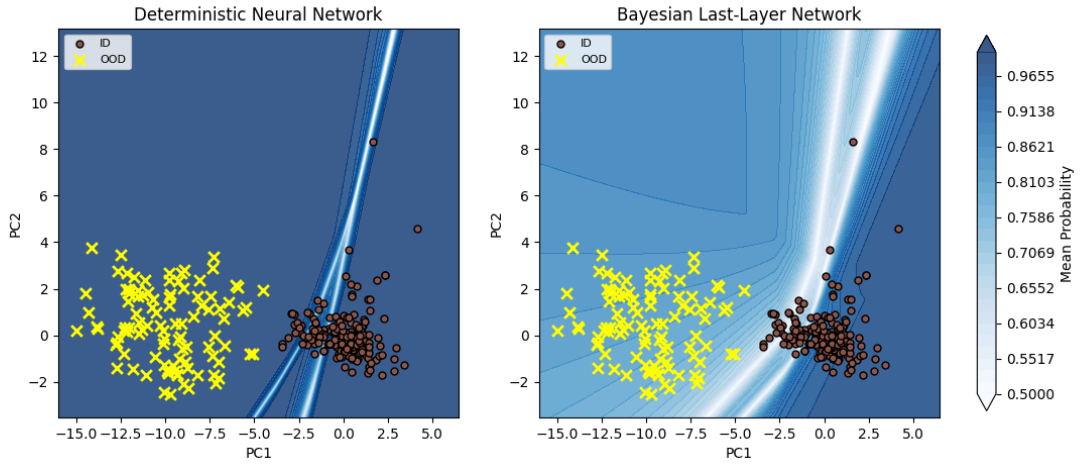

Figure 2: Decision boundaries and predictive mean probabilities of the deterministic and BLL network on ID and OOD data projected via PCA.

Moreover, the categorical column was one hot encoded to ensure suitability with the neural networks. Due to class imbalance in the target labels, the dataloader object was configured with class weights responsible for up-weighing underrepresented categories. Given the restricted dataset size, the models were initialized with an input layer of size 11 followed by 2 hidden layers of sizes 8 and 4, and an output layer of size 3. The data was split into training and test set using a 80/20 split. In addition, 100 instances of OOD data, sampled from a scaled uniform distribution, were added to the test set; see (9). The Figure A.1 showcases how the OOD data is distributed in comparison to ID data in the feature extractor.

$$X^{\text{OOD}} \overset{\text{iid}}{\sim} 10 \cdot \text{Uniform}(-1, 1), \quad X^{\text{OOD}} \in \mathbb{R}^{100\times 11} \tag{9}$$

Similar to the previous task, both models were trained using a batch size of 32 and the ADAM optimizer with a learning rate of 0.001. For this task, the CE was used as the likelihood term in the ELBO. The deterministic model, as well as the shared base model, was trained for 190 epochs, while the BLL was trained separately for 100 epochs to avoid overfitting, which was observed with longer training durations. The hyperparameters were manually selected based on validation performance, resulting in prior variances of -0.3 for the weights and biases, and 0.55 for the prior standard deviation used in the KL divergence term.

Proceeding with model evaluation on the test set, Figure 2 illustrates the decision boundaries of both networks, with the predictive mean probability visualized using blue shading. The two axes represent the two components that explain the highest variability of the data, obtained via principal component analysis (PCA). It can be seen that for the OOD data marked in yellow, the deterministic network assigned a high mean probability near 1. In contrast, the BLL network is assigned a lower predicted probability around 0.7, highlighted by the brighter blue shade.

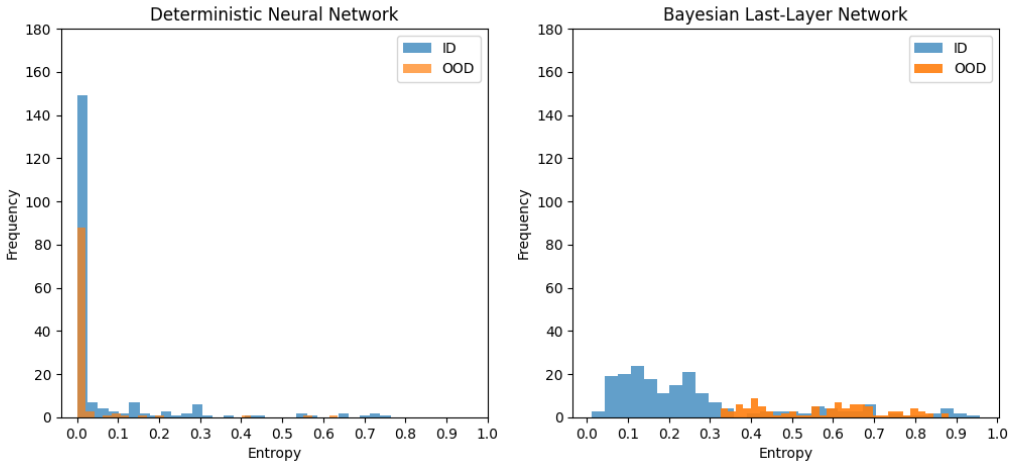

Figure 3: Shannon Entropy distributions of predictions for ID and OOD data for the deterministic and BLL network.

Moreover, Figure 3 displays the Shannon entropy of both models for ID and OOD data in a histogram. The deterministic network's entropy is concentrated near zero for both ID and OOD data. In contrast, the BLL network depicts a higher entropy for OOD data focused around 0.4 to 0.8.

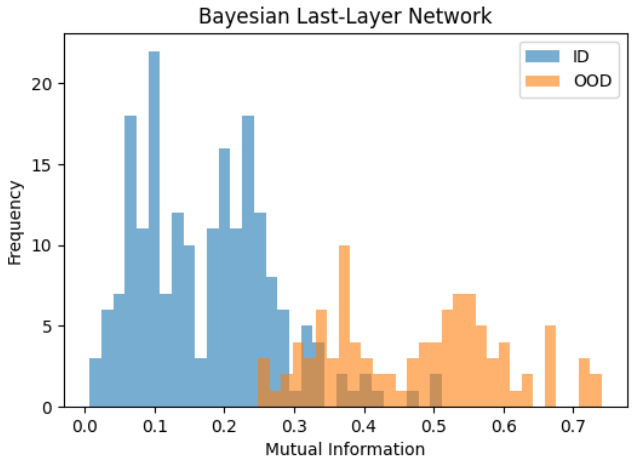

Figure 4: Mutual information distributions for ID and OOD data from the BLL network.

This behavior of the BLL network can additionally be observed in Figure 4, where a visible separation in MI between ID and OOD can be seen. Lastly, the ROC curve for OOD detection using the Shannon entropy is shown in Figure A.2, where the BLL network achieves an AUROC of 0.85.

5 Discussion

The outcome of the experiment provides insight into the impact of different neural network architectures in the context of uncertainty estimation. The analysis supports the introduced notion that deploying partially Bayesian neural networks on OOD data settings allows for improved uncertainty calibration. As observed in the regression task, introducing a Bayesian last-layer led to increased standard deviation and PI coverage on OOD data, indicating a better ability to capture epistemic uncertainty.
In contrast, the deterministic network maintained a constant confidence band, reflecting its restricted ability to only capture aleatoric uncertainty. In the classification task, the BLL network produced a lower predicted mean probability as well as a higher Shannon entropy on OOD data. This suggests an improved ability to capture uncertainty, in contrast to the deterministic network assigning a predicted mean probability of 1, undermining its overconfidence.

In line with the hypothesis, the results support the benefit of introducing stochasticity in the last layer of a neural network. However, the performance did not fully meet all expectations.
More specifically, a clearer separation of entropy between ID and OOD observations was anticipated for the BLL network, with an expected entropy near 1 for OOD data. Furthermore, the BLL network ideally would have assigned a mean probability of 0.5 to all observations that were either OOD or not in the dataset.

In summary, the results align with the idea of Kristiadi et al. (2020), however they highlight certain limitations with the implementation of the BLL network.

The hyperparameter selection is crucial for the results. Due to the fact that they were obtained through manually defined grid search, the limitation lies in the grid size and computational resources to evaluate numerous possibilities.
Furthermore, as mentioned in the methods chapter, solely including stochasticity in the last layer ignores uncertainty introduced by the previous layers, affecting the uncertainty estimation capability of the network.
Finally, the limited dataset size and choice may affect the generalizability of the results, requiring a bigger size to obtain more reliable results.

6 Conclusion and Outlook

This seminar thesis aimed to empirically investigate the effect of utilizing partially stochastic neural networks, in particular Bayesian last-layer networks, in the context of uncertainty estimation in out-of-distribution settings. The conducted experiment confirms that by introducing stochasticity in the last layer, one achieves improved uncertainty estimation and lower confidence on OOD data. Both the regression task on simulated data and the classification task using real-world data, portrayed an improved behavior, measured by a lower entropy, a clearer separation of MI metrics, and predicted mean probability on OOD data. In contrast, the deterministic network remained overconfident on all data.

This work contributes to current research by providing a validation of the Bayesian last layer method proposed by Kristiadi et al. (2020). By comparing a deterministic neural network to a partially stochastic neural network with a Bayesian last layer, this thesis, serves as a benchmark, allowing insights into theoretical properties as well as practical advantages and limitations.

As previously mentioned, the implementation of a Bayesian last-layer network using Bayes by Backprop was subject to certain limitations. Therefore, further research should explore possibilities of an improved hyperparameter selection in addition to evaluating the method on larger datasets, improving generalizability.
Furthermore, improved methods such as SNGP should be considered to explore improved uncertainty estimation, as well as as different approximation methods for the posterior.

In summary, this thesis validates the practical benefits of the Bayesian last layer approach, while also highlighting its current limitations and setting a clear direction for the future research.

A Appendix

Table A.1: Activation functions for hidden and output layers

Layer Type	Activation	Formula
Hidden	ReLU	$\max(0, x)$
	Sigmoid	$\frac{1}{1+e^{-x}}$
Output	Linear	x
	Softmax (multiclass)	$\frac{\exp(x_i)}{\sum_{j=1}^{k}\exp(x_j)}$

Table A.2: Uncertainty measures

Measure	Formula
Shannon entropy	$H[Y \mid x] = -\sum_y p(y \mid x)\, \log p(y \mid x)$
Expected entropy	$\mathbb{E}_{w\sim p(w\mid D)}\big[H[Y \mid x, w]\big] = -\int p(w \mid D) \sum_y p(y \mid x, w) \log p(y \mid x, w)\, dw$
Mutual information	$\mathrm{MI}[Y, w \mid x, D] = H[Y \mid x] - \mathbb{E}_{w\sim p(w\mid D)}\big[H[Y \mid x, w]\big]$

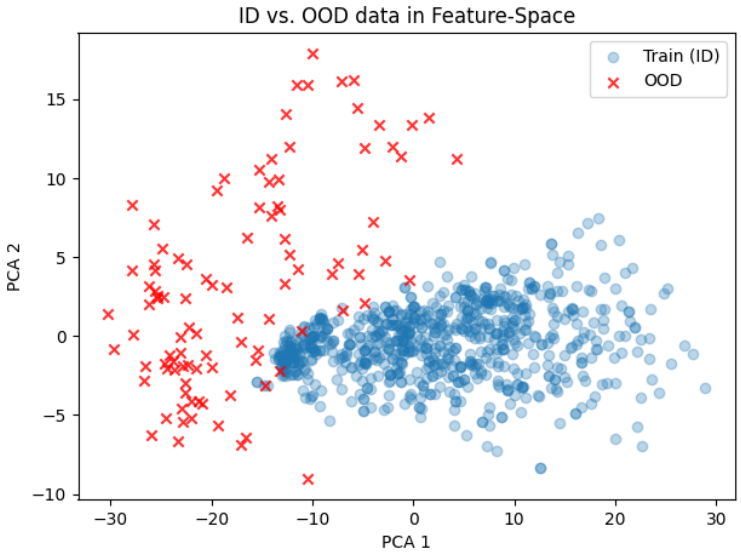

Figure A.1: PCA projection of ID and generated OOD samples in feature space of the base network of the BLL.

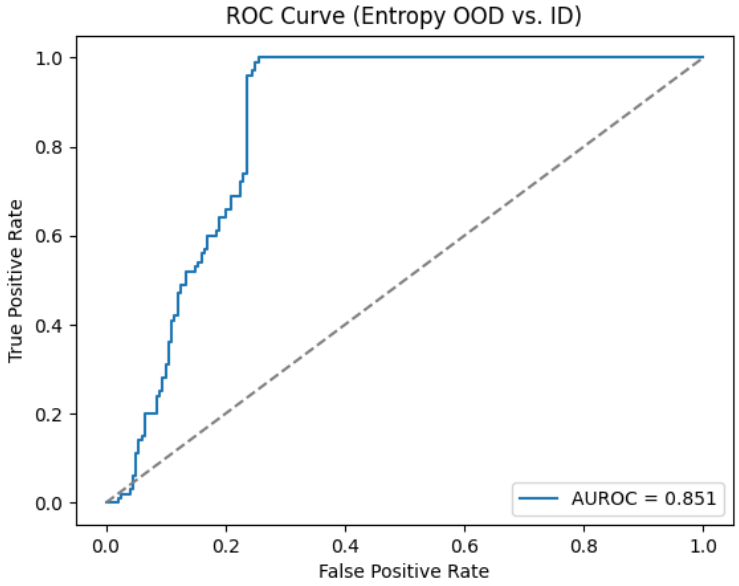

Figure A.2: ROC curve of BLL network for OOD and ID data based on Shannon entropy (AUROC = 0.851).

B Electronic appendix

Data, code and figures are provided in electronic format. Data for diabetes classification dataset is available on :

https://www.kaggle.com/datasets/marshalpatel3558/diabetes-prediction-dataset-legit-dataset

Code is available on:

https://github.com/Kingmopser/ProbabilisticML

References

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M. and Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, Association for Computing Machinery, New York, NY, USA, p. 1721–1730.
URL: *https://doi.org/10.1145/2783258.2788613*

Foong, A. Y. K., Li, Y., Hernández-Lobato, J. M. and Turner, R. E. (2019). 'in-between' uncertainty in bayesian neural networks.
URL: *https://arxiv.org/abs/1906.11537*

Girshick, R. (2015). Fast r-cnn.
URL: *https://arxiv.org/abs/1504.08083*

Hein, M., Andriushchenko, M. and Bitterwolf, J. (2019). Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem.
URL: *https://arxiv.org/abs/1812.05720*

Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods, *Machine Learning* **110**(3): 457–506.
URL: *http://dx.doi.org/10.1007/s10994-021-05946-3*

Korotin, A., Egiazarian, V., Asadulaev, A., Safin, A. and Burnaev, E. (2020). Wasserstein-2 generative networks.
URL: *https://arxiv.org/abs/1909.13082*

Krause, A. and Hübotter, J. (2025). Probabilistic artificial intelligence.
URL: *https://arxiv.org/abs/2502.05244*

Kristiadi, A., Hein, M. and Hennig, P. (2020). Being bayesian, even just a bit, fixes overconfidence in ReLU networks, *in* H. D. III and A. Singh (eds), *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119 of *Proceedings of Machine Learning Research*, PMLR, pp. 5436–5446.
URL: *https://proceedings.mlr.press/v119/kristiadi20a.html*

Murphy, K. P. (2022). *Probabilistic Machine Learning: An introduction*, MIT Press.
URL: *http://probml.github.io/book1*

Murphy, K. P. (2023). *Probabilistic Machine Learning: Advanced Topics*, MIT Press.
URL: *http://probml.github.io/book2*

Riquelme, C., Tucker, G. and Snoek, J. (2018). Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling.
URL: *https://arxiv.org/abs/1802.09127*

Seoni, S., Vicnesh, J., Salvi, M., Barua, P. D., Molinari, F. and Acharya, U. (2023). Application of uncertainty quantification to artificial intelligence in healthcare: A review of last decade (2013–2023), *Computers in Biology and Medicine* **165**: 107441.

Sharma, M., Farquhar, S., Nalisnick, E. and Rainforth, T. (2023). Do bayesian neural networks need to be fully stochastic?
URL: *https://arxiv.org/abs/2211.06291*

Wimmer, L., Sale, Y., Hofman, P., Bischl, B. and Hüllermeier, E. (2023). Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures?
URL: *https://arxiv.org/abs/2209.03302*

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.

Location, date

Freising, 03.07.2025

Bakir Chaban

Name