ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Revisiting Neural Activation Coverage for Uncertainty Estimation

Benedikt Franke[†1], Nils Förster[1], Frank Köster[1], Asja Fischer[2], Markus Lange[1] and Arne Raulf[1]
[†] Corresponding author contact: benedikt.franke@dlr.de.

[1]DLR Institute for AI Safety and Security, Ulm - Germany
[2]Ruhr University Bochum, Bochum, Germany

Abstract. Neuron activation coverage (NAC) is a recently-proposed technique for out-of-distribution detection and generalization. We build upon this promising foundation and extend the method to work as an uncertainty estimation technique for already-trained artificial neural networks in the domain of regression. Our experiments confirm NAC uncertainty scores to be more meaningful than other techniques, e.g. Monte-Carlo Dropout.

1 Introduction

For safety-critical applications, uncertainty estimation (UE) of artificial neural networks (ANNs) is an important research area to counter-act overconfidence of wrong decisions and ensure trustworthiness of the predictions of the ANN. However, a lot of UE techniques require intervention at training time, modifying the architecture of the ANN to explicitly calculate uncertainty measures [1]. This makes re-using pre-trained models challenging and may even impact the final performance of the model. Wrapper-like UE methods that can be applied to a trained ANN without re-training or finetuning the neural network are only sparsely explored in the literature [2, 3, 4]. To this end, neuron activation coverage (NAC), a technique to calculate an uncertainty measure based on the activation pattern of a trained convolutional neural network (CNN), was recently introduced [5]. In this work, we extend the method to work as an uncertainty estimator for regression, and verify the results by comparison with other UE methods. Our contributions can be summarized as follows:

1. We extend the NAC methodology to regression by proposing a new objective function.
2. We confirm the competitiveness of NAC against other uncertainty estimations techniques like Monte-Carlo Dropout (MC dropout) [1] by direct comparison.
3. We publicly release our code, including an optimized, easy-to-reuse, single-file adaptation[1] of NAC for PyTorch [6].

[1]`https://github.com/DLR-KI/nac-uncertainty-regression`

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

2 Related Work

Uncertainty estimation in ANNs has become a crucial area of research, especially for applications requiring reliable confidence measures. One kind of uncertainty estimation focuses on calculating the uncertainty of ANNs post-training, so that no modifications to the training protocol or network architecture are necessary. These include approaches such as deep ensembles [7], which aggregate predictions from multiple independently trained networks, as well as methods based on test-time data augmentation [4], which evaluate prediction variability by feeding augmented inputs through the fixed network. Furthermore, calibration methods adjust output probabilities to better reflect uncertainty without altering the network weights [8].

Another kind of uncertainty estimation methods operates during training, explicitly incorporating uncertainty modeling into the learning process [9, 10]. One representative of this concept is MC dropout [1], where dropout is applied at both training and inference time to approximate Bayesian inference. By sampling multiple stochastic forward passes through the network at test time, MC dropout generates a distribution over predictions, providing an uncertainty estimate directly linked to the model's learned parameters.

3 Extending Neural Activation Coverage for Uncertainty Estimation

3.1 Background

We briefly restate the definition of NAC as introduced in reference [5] with minor deviations of notation to clarify our adaptations. Interested readers are referred to reference [5] for the more exhaustive introduction. NAC calculates *activation states* of selected neurons in an ANN by backpropagating a 'pseudo-loss' L. This pseudo loss is given by the KL divergence between a uniform output vector u and the network output as softmax scores p, thus we have $L(p) = D_{\mathrm{KL}}(u||p)$.

Using the output z of one neuron of some intermediate layer and the pseudo-loss $L(p)$, the activation state $\hat{z}$ is defined in Equation (1). It is clamped between `0` and `1` by a sigmoid function σ.

$$\hat{z} = \sigma(z \odot \frac{\partial L(p)}{\partial z}) \tag{1}$$

During calibration time, an approximate probability density function κ_X^i is built up based on the computed $\hat{z}$-values of known in-distribution (ID) data X for each neuron i in the chosen layer(s). This probability density function κ_X^i is approximated as an histogram in actual implementation. Using the parameter $r > 0$, defining the upper limit for each histogram bin, the NAC-function is then defined as

$$\Phi_X^i(\hat{z}_i; r) = \frac{1}{r}\min(\kappa_X^i(\hat{z}_i), r)\,. \tag{2}$$

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Intuitively, $\Phi_X^i(\hat{z}_i; r)$ can be interpreted as a "recognition score", that increases the more often a neuron activation score has been observed during calibration phase. For calculating an ANN's uncertainty, we consider it as a set of layers M. For each $m \in M$, the uncertainty scores of it's neurons are averaged. To arrive at a final uncertainty score, the individual layer scores are summed. For this, we look up the respective activation scores $\hat{z}_i$, according to Equation (1), and arrive at a set of activation states $\hat{Z} = \{\hat{z_1}, \hat{z_2}, \ldots, \hat{z_n}\}$, with n being the number of neurons in the network [5]. We use the neuron-wise density functions Φ_X^i for each neuron $i \in m$, to arrive at the final score

$$S(M, X) = \sum_{m \in M} \frac{1}{|m|} \sum_{i \in m} \Phi_X^i(\hat{z}_i; r)\,. \tag{3}$$

As we are interested in computing uncertainty scores, we take the inverse of the NAC score $U_{\mathrm{NAC}} = S(M, X)^{-1}$, which is defined as long as $M, X \neq \emptyset$.

3.2 NAC for Regression

For regression problems, computing the KL-Divergence $L(p)$ is not possible, since the output of regression ANNs is one (or more) scalar value(s), and not a probability vector. However, since the uniform class probability vector $u = (1/C, \ldots, 1/C)^{\mathrm{T}}$ for C classes is also the mean output given an independent and identically distributed (i.i.d.) dataset, we propose a new pseudo-loss for regression problems. Let $\bar{y}_X$ denote the mean output vector over some ID calibration dataset X. For a given vector p of predictions of the network we utilize the *Mahalanobis* distance [11] d_{m} with $L_X(p) = d_{\mathrm{m}}(p, \bar{y}_X)$.

With this new definition for $L_X(p)$, NAC can now be efficiently used for computing the uncertainty of regression problems. In Chapter 4, we empirically evaluate this new definition.

4 Experiments

To evaluate the suitability of NAC towards assessing uncertainty specifically for detecting out-of-distribution (OoD) data, we train a 3-layer multi-layer Perceptron (MLP) on ten regression datasets of the UCI repository [12]. The network has 128 neurons per hidden layer and uses SELU activations [13].

4.1 Out-of-Distribution Detection

We generate synthetic artificial data by projecting each data point outside of the original distribution by adding noise drawn from a $\mathcal{N}(4\sigma, \frac{\sigma}{2})$ distribution where σ is the standard deviation vector of the original dataset, i.e., for a dataset originally distributed as $\mathcal{N}(\mu, \sigma)$. Our OoD data is distributed as $\mathcal{N}(\mu + 4\sigma, \frac{3\sigma}{2})$ and should therefore be sufficiently distinguishable from the original data.

To verify our approach, we compare NAC to established methods, such as ensembling, and MC dropout. For ensembling, we train ten networks simultaneously with bootstrapping data and take the standard deviation of the predictions

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

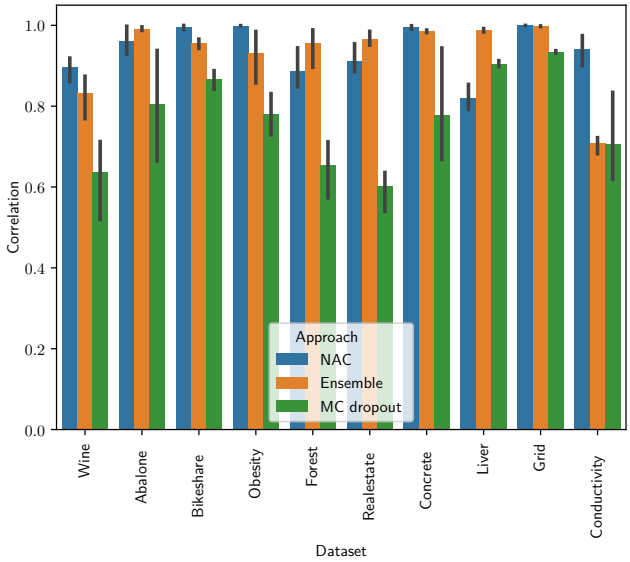

Fig. 1: **Comparing uncertainty approaches for OoD detection.** Correlation of uncertainty value and OoD-Label on ten UCI regression datasets. NAC adaptations (ours) achieve the best correlation in six out of ten experiments. Error bars represent the 95% confidence interval.

as measure of uncertainty. For MC dropout, we insert a dropout layer after each linear layer and compute the uncertainty as the standard deviation of ten forward passes. As NAC introduces additional hyperparameters, we split off 10% of the training data and generate an equal amount of OoD-data for a quick hyperparameter sweep in each experiment after training the ANN. We compare the correlation between the uncertainty value and a binary marker for artificially created OoD-data (`0` for ID, `1` for OoD) across ten UCI-datasets for regression. Our results are visualized in Figure 1.

As depicted, NAC's uncertainty correlates best with the OoD attribute in six out of ten experiments, which makes it the strongest uncertainty estimation technique of the three compared approaches. In the four other cases it produces mostly comparative results to the other uncertainty estimation techniques. These results show that NAC is well-suited to compute uncertainty values for OoD detection, even in the case of regression.

4.2 Uncertainty Calibration

As uncertainty can stem from a multitude of causes [14], we want to find out if NAC scores are influenced by errors on the ID data. We repeat the previous comparison setup, however we do not generate any artificial data, but compute the correlation between the mean squared error (MSE) of the prediction on the

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

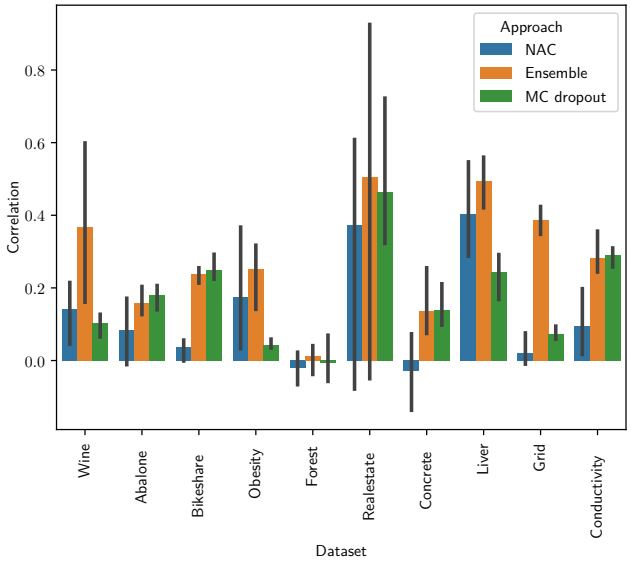

Fig. 2: **Comparing uncertainty approaches for ID error detection.** Correlation of uncertainty value and per-sample MSE on ten UCI regression datasets. NAC adaptations (ours) achieve the smallest correlation in seven out of ten experiments. Error bars represent the 95% confidence interval.

test split and the uncertainty value. Results are depicted in Figure 2. In stark contrast to Figure 1, we see that NAC scores do not have a high predictive power for errors on ID data, achieving the lowest correlation value of the three methods in seven out of ten cases. This indicates a high calibration towards OoD-driven uncertainty for NAC-scores.

5 Conclusion and Future Work

In this work, we extended NAC to regression and compared it with two other uncertainty techniques across ten UCI datasets. Our results show that NAC is not only well-suited to compute uncertainty for OoD detection. In contrast, the approach tends to not be influenced by ID prediction errors, enabling practitioners to efficiently compute uncertainty for OoD detection.

We observed that NAC is easier on computational resources when comparing the presented approaches, as both MC dropout and ensembling need multiple forward passes, ensembling even requiring multiple trained models. Like ensembling, it does not need any modification to trained ANNs, but can work on a single ANN. This also applies to MC dropout only if the trained network already contains dropout layers, which isn't the case in many modern applications of ANNs, e.g. modern deep CNNs. In future work, we aim to extend NAC even

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

further, e.g. towards Object Detection, which can be seen as a combination of classification and regression.

References

[1] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[2] Philipp Oberdiek, Matthias Rottmann, and Hanno Gottschalk. *Classification Uncertainty of Deep Neural Networks Based on Gradient Information: 8th IAPR TC3 Workshop, ANNPR 2018, Siena, Italy, September 19–21, 2018, Proceedings*, pages 113–125. 08 2018.

[3] Tiago Ramalho and Miguel Miranda. Density estimation in representation space to predict model uncertainty. In *International Workshop on Engineering Dependable and Secure Machine Learning Systems*, pages 84–96. Springer, 2020.

[4] Murat Seckin Ayhan and Philipp Berens. Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. In *Medical Imaging with Deep Learning*, 2018.

[5] Yibing Liu, Chris Xing Tian, Haoliang Li, Lei Ma, and Shiqi Wang. Neuron activation coverage: Rethinking out-of-distribution detection and generalization. In *The Twelfth International Conference on Learning Representations*, 2024.

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library, 2019. `https://arxiv.org/abs/1912.01703`.

[7] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[8] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[9] Ankush Ganguly and Samuel WF Earp. An introduction to variational inference. *arXiv preprint arXiv:2108.13083*, 2021.

[10] Ethan Goan and Clinton Fookes. *Bayesian Neural Networks: An Introduction and Survey*, page 45–87. Springer International Publishing, 2020.

[11] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Sankhyā: The Indian Journal of Statistics, Series A (2008-)*, 80:S1–S7, 2018.

[12] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The uci machine learning repository, 2023. `https://archive.ics.uci.edu`.

[13] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *Advances in neural information processing systems*, 30, 2017.

[14] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5580–5590, Red Hook, NY, USA, 2017. Curran Associates Inc.