A review: Training independent subnetworks for robust prediction

Milan Kalkenings
Wirtschaftsinformatik und Maschinelles Lernen
Universität Hildesheim
Samelsonplatz, 31141 Hildesheim
milan.kalkenings@uni-hildesheim.de

Abstract

Fusing predictions from a diverse set of estimators leads to better generalization and the ability to estimate prediction uncertainty. This work reviews a research paper that proposes training a light weighted set of versatile estimators with decorrelated errors within a single neural network. The reviewed work succeeds in proposing a novel, versatile, and powerful approach to reach that goal. Throughout this work, mathematical formulations, experimental setups, ways of reasoning, and interpretations are corrected, and thus implicitly explained.

1 Introduction

This work reviews "Training Independent Subnetworks for Robust Prediction" (Havasi et al., 2021) that addresses the research question of whether it is possible to train an implicit ensemble of diverse subnetworks within a single neural network. One main advantage of having predictions from versatile ensemble members is that their variance can be mapped to a prediction uncertainty score that can guide decision making (Nado et al., 2022). The subsequent section outlines ensembling techniques for deep neural networks proposed by related works to set (Havasi et al., 2021) in a wider context. Afterwards, the reviewed article is briefly summarized and discussed, before the main review findings are concluded.

2 Related Work

Dropout networks (Srivastava et al., 2014) are implicit ensembles, averaged over the subnetworks used in the individual update steps that solely provide a point prediction (Hara et al, 2016). Monte Carlo Dropout (Gal and Gharamani, 2016) drop activations in inference and thus provide as many member predictions as forward passes are performed. Deep Ensemble (Lakshminarayanan et al., 2017) trains neural entworks from different parameter initializations and leads to diverse members. Multiheaded methods like TreeNet (Lee et al., 2015) ease the complexity by sharing the parameters of initial layers among all members. Likewise, BatchEnsemble (Wen et al., 2020) forms ensemble members by adding individual parameter extensions to shared base parameters. Other techniques like Snapshot Ensembles (Huang et al., 2017) create ensembles along one training trajectory.

3 Summary

The introduction section of (Havasi et al., 2021) states that prediction uncertainty estimation is vital for decision-making and identifies ensembling neural networks as a popular method to achieve them while demanding multiple forward passes. Consequently, the authors propose the Multi-Input Multi-Output (MIMO) configuration that trains an implicit ensemble, evaluated within a single forward

Preprint. Under review.

pass of one neural network to reduce the time and space complexity. MIMO ensembles demand M concatenated inputs and provide M predictions using individual output heads. During training, M independent inputs are fed into the network, whereby output head m learns to predict the correct label for input m. In inference, the same input is fed M times into the ensemble, and the head outputs are averaged to achieve an ensemble prediction. Neural networks can be transformed into MIMO ensembles with negligible increase in time and space complextity. (Havasi et al., 2021) show that the non-shared parameters of MIMO members converge in disconnected accuracy modes, resulting in high disagreement among the MIMO member predictions in comparison to Naive Multihead, TreeNet, and BatchEnsemble baselines, recreating that of a way heavier Deep Ensemble. Experiments reveal, that neurons in MIMO ensembles form implicit subnetworks: independent connections between the input-output pairs that can fully leverage the network capacity if M is chosen appropriately. (Havasi et al., 2021) claim that repeating the same batch within an update step can increase the ensemble performance, and that feeding the same input into multiple input heads can relax the subnetwork independence, allowing higher M without exceeding the network capacity. MIMO ensembles are benchmarked against a deterministic neural network, Monte Carlo Dropout, BatchEnsemble, Naive Multihead, TreeNet, and Deep Ensemble on CIFAR10, CIFAR100 (Krizhevsky and Hinton, 2009) and ImageNet (Deng et al., 2009) image classification, whereby all models were fomred based on members of the ResNet (He et al., 2016) family. The loss, the accuracy and the expected calibration error (ECE) (Naeini et al., 2015), achieved on the test sets and perturbated versions of them, are compared to the evaluation runtime of the models. Thereby, ECE is used to determine the gap between the model confidence and its reliability. The benchmarks show that the proposed method achieves accurate and reliable results considering its low prediction time. (Havasi et al., 2021) emphasize that their approach is easy to implement, computationally efficient, robust, and compatible with many further training techniques. Additional information about the implementation of the proposed method is issued in the form of a pseudocode appendix, an interactive notebook, and open source code in a GitHub repository. (Havasi et al., 2021)

4 Discussion

This section discusses the quality of the reviewed article. The introduction motivates and briefly describes the proposed method before ending in a motivating enumeration of the main contributions.

Section 2 explains the proposed method in details, thus having overlaps with the introduction, and demonstrates it on a toy regression data set. Thereby, the authors define $\hat{f}_M$ to be a MIMO ensemble with M members, so that $\hat{f}_M(x', ..., x')$ is the prediction of a whole ensemble, and $\bar{f}_M(x', ..., x')$ is the averaged prediction of multiple ensembles across a number of runs. The authors deconstruct the expected mean squared error into the (squared) bias error $\mathbb{E}_{(x',y')\in\mathbb{X}_{test}}[(f_M(x', ..., x') - y')^2]$ and the variance error $\mathbb{E}_{(x',y')\in\mathbb{X}_{test}}[\mathbb{E}_{\mathbb{X}}[(f_M(x', ..., x') - \hat{f}_M(x', ..., x'))^2]]$. The authors state that a large bias error reveals weak ensemble members, and that a low variance error indicates large diversity in the member predictions. These conclusions are not valid, since it is not possible to deduct information about the individual members by investigating the performance on the ensemble level. For the given model and data, the experiment actually provides two findings that were not addressed by the authors: The correlation of M and the bias error shows, that the average prediction of multiple MIMO ensembles $\bar{f}_M(x', ..., x')$ diverges from the target with increasing M, probably due to exceeding the network capacity. The negative correlation of M and the variance error on the other hand reveals that the predictions of MIMO ensembles of the same size become more and more similar as M increases. This finding indicates that larger MIMO ensembles are less sensitive to small changes in the training data. (Ganaie et al., 2022) define the variance error as the average variance between the member predictions, and the bias error as their average difference to the ensemble prediction to investigate the performance on the member level. That methodology could achieve actual insights about MIMO member predictions.

Section 3 analysis MIMO ensembles and their training. Despite its poor performance in the benchmarks in section 4, a Naive Multihead baseline is used in subsection 3.1. A comparison with further baselines like TreeNet and Deep Ensemble would provide less biased results, that could embed the peculiarities of the proposed method in a wider picture. Figure 3 captures the subsections results, whereby the legend is divided among the two middle-right and the two right subplots. The figure could have been improved by adding an additional subplot solely containing one comprehensive

legend and deleting the existing legends. Nevertheless, this subsection shows that the non-shared MIMO weights converge in disconnected accuracy modes and highly disagree in their predictions.

Subsection 3.2 further analyzes the predictions made by the MIMO subnetworks. Firstly, (Havasi et al., 2021) train a MIMO ensemble and iteratively checkpoint its member predictions after a number of iterations. This experiment doesn't provide valuable information, since the six most right subplots of figure 3 already demonstrate the disagreement of the MIMO head predictions on the same test set. The only additional information from this experiment is that the predictions of the MIMO members are more similar before training the ensemble. This is a trivial fact, since all network parameters were randomly initialized, resulting in random but similar member predictions. In the second experiment, the authors compare the diversity of the member predictions of the proposed method against multiple baselines. The authors define the distance scores $D_{disagreement}(P_1, P_2)$ and $D_{KL}(P_1, P_2)$ for predictive distributions, i.e. class score vectors P_1 and P_2. The authors claim that $D_{KL}(P_1, P_2)$ calculates the Kullback-Leibler divergence (Kullback and Leibler, 1951) between P_1 and P_2 and define it as displayed in Equation 1. However, the definition is flawed, because it forms the expected value over P_1 instead of the vector entries y, and the element wise differences have to be scaled by P_1 to actually achieve the Kullback-Leibler divergence as displayed in Equation 2. The scores were averaged over all member pairs, rounded to three digits after the decimal point, and displayed in figure 4, containing a number of questionable results: Firstly, it is unlikely that the proposed method and the Deep Ensemble baseline achieve the exact same results. Even more unlikely is that the Naive Multihead baseline achieves a value of 0.000 for both scores. For example, a minimum of 15 disagreements is necessary to achieve a rounded $D_{disagreement}$ of 0.001, since the test set contains 10.000 data points (Krizhevsky and Hinton, 2009). The same baseline formed from a smaller neural network achieves a way higher head disagreement on the same test set in figure 3, making these results even more unlikely.

$$D_{KL}(P_1, P_2) = \mathbb{E}_{P_1}[logP_1(y) - logP_2(y)] \tag{1}$$

$$D_{KL}^{correct}(P_1, P_2) = \mathbb{E}_y[P_1(y)(logP_1(y) - logP_2(y))] \tag{2}$$

Subsection 3.3 investigates to which degree shared pre-activations in MIMO ensembles form independent subnetworks. For a MIMO model with $M = 3$, the authors define the conditional variance of a pre-activation a_i with respect to x_1 as shown in Equation 3. Confusingly, the textual description states that it can be estimated as shown in Equation 4. Subsequently, the authors state that a pre-activation belongs to subnetwork j, if its conditional variance with respect to x_j, i.e. the degree to which x_j impacts a_i, is non-zero. However, Equation 4 falls back to averaging over multiple $Var_{x_2,x_3}(a_i(x_1, x_2, x_3))$ with fixed x_1. Thereby, the impact of x_1 on a_i can't be measured by this procedure, but instead the impact of varying x_2 and x_3 is captured. Consequently, a non-zero output of Equation 4 only indicates, that a_i depends on either x_2, x_3, or both of them. That indicates that a_i belongs to at least one of the two subnetworks corresponding to x_2 or x_3, and it can not be deducted if a_i belongs to the first subnetwork as well. A non-zero output of Equation 3 on the other hand in fact reveals that a_i depends on x_1. These findings can be applied to the provided definitions for MIMO ensembles of size $M = 2$ as well. The figures show that many pre-activations have a conditional variance of (highly) non-zero with respect to just one input, and a (close-to) zero conditional variance with respect to the other inputs. This shows, that many neurons in fact belong to one subnetwork only. Nevertheless, it becomes evident that there are still some outliers that consequently highly belong to more than one subnetwork.

$$Var(a_i|x_2, x_3) = \mathbb{E}_{x_2,x_3}[Var_{x_1}(a_i(x_1, x_2, x_3))] \tag{3}$$

$$Var(a_i|x_2, x_3) = \mathbb{E}_{x_1}[Var_{x_2,x_3}(a_i(x_1, x_2, x_3))] \tag{4}$$

Subsection 3.4 and appendix C demonstrate that using the proposed method can help to leverage the capacity of a neural network, resulting in a better trade off between space complexity and prediction accuracy.

Subsection 3.5 introduces two hyperparameters. The introduction of the input repetition probability is motivated by stating that feeding the same input to multiple subnetworks relaxes their independence

and enables low-capacity networks to be transformed into a MIMO ensemble. The motivation of the batch repetition hyperparameter on the other hand remains unclear. The authors argue that feeding the concatenation of multiple repetitions of the same batch into a MIMO network leads to a similar result to feeding multiple posterior samples of the same data point into a stochastic model. The problem with this argumentation is that MIMO ensembles are deterministic if not formed from a non-deterministic network. Thus, repeating the same input doesn't reduce their gradient noise as it is the case in stochastic models. However, the results show that batch repetition can improve the performance, but neither the subsection nor the respective figure caption explains which architecture and which data set were used to achieve them. It remains unclear why batch repetition improves the results, but it might be due to stochastic building blocks in the utilized architecture.

Section 4 benchmarks the proposed method against a number of baselines and displays the results in three tables. ECE was used to meter the calibration of the compared models. It bins all predictions by their confidence score for the ground truth class into a fixed number of equally ranged bins B, and is then calculated as the absolute difference between the bin accuracy and the average bin confidence for the ground truth class (Naeini et al., 2015). (Nixon et al., 2019) argue that ECE is not a suitable evaluation metric for multi-class classification problems like those at hand, because the confidence scores for the non-ground-truth classes are not taken into account, and propose averaging over all classes. Within the first table, the authors state to use a "Dropout" baseline model, while they use a "Monte Carlo Dropout" baseline in the second table. Most likely the authors refer to the Monte Carlo Dropout method in both cases, since they mention to use the MC-Dropout baseline within the subsection body. Using three different names can lead to confusion, especially since a neural network with dropout layers is sometimes interpreted as an implicit ensemble itself (Hara et al., 2016). Assuming that the authors indeed used Monte Carlo Dropout, a second and more important issue arises: In all tables, only one forward pass was performed with it. Since Monte Carlo Dropout refers to building an implicit ensemble across multiple forward passes by dropping out different activations in inference, performing just one forward pass only results in a non-deterministic point prediction (Gal and Gharamani, 2016). Furthermore, benchmarking on more tasks than image classification and forming the models from further architectures than those of the ResNet family would provide valuable insights into the applicability of MIMO ensembles.

5 Conclusion

The related works section of this work reveals that (Havasi et al., 2021) was well motivated, since the high demand of time and space in related approaches can represent a bottleneck in many applications. (Havasi et al., 2021) follows a clear concept and shows that the proposed method can train a diverse set of estimators that generalize well and provide a prediction uncertainty estimation. However, some results seem unlikely, one experiment doesn't provide any valuable insights, and the proposed method was merely benchmarked in highly related settings. Furthermore, using Monte Carlo Dropout with only one forward pass, the gap between equations and their descriptions and the interpretation of the results in subsection 2.1 reveal major theoretical misconceptions.

References

Deng, J., Dong, W. Socher, R., Li, L. J., Li, K., Fei-Fei, L. (2009):
"ImageNet: A large-scale hierarchical image database",
2009 IEEE Conference on Computer Vision and Pattern Recognition,
Miami, USA, 2009, pp. 248-255.

Gal, Y., Gharamani, Z. (2016):
"Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning",
Proceedings of the 33rd International Conference on International Conference on Machine Learning, vol. 48, pp. 1050-1059,
New York, USA, 2016.

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., Suganthan, P. N. (2022):
"Ensemble deep learning: A review",
Engineering Applications of Artificial Intelligence, vol. 115, p. 105151.

Hara, K. Saitoh, D., Shouno, H. (2016):

"Analysis of dropout learning regarded as ensemble learning",
Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks,
Barcelona, Spain, 2016.

Havasi, M., Jenatton, R., Fort, S., Liu, J., Snoek, J. R., Lakshminarayanan, B., Dai, A. M., Tran, D. (2021):
"Training independent subnetworks for robust prediction",
International Conference on Learning Representations,
Vienna, Austria, 2021.

He, K., Zhang, X., Ren, S., Sun, J. (2016):
"Deep Residual Learning for Image Recognition",
Conference on Computer Vision and Pattern Recognition (CVPR),
Las Vegas, USA, 2016, pp. 770-778.

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., Weinberger, K., Q. (2017):
"Snapshot Ensembles: Train 1, Get M for Free",
International Conference on Learning Representations.

Krizhevsky, A. Hinton, G. (2009):
"Learning Multiple Layers of Features from Tiny Images",

Kullback, S., Leibler, R. A., (1951):
"On information and sufficiency."
The annals of mathematical statistics, vol. 22 pp.79-86.

Lakshminarayanan, B., Pritzel, A., Blundell, C. (2017):
"Simple and scalable predictive uncertainty estimation using deep ensembles",
Advances in neural information processing systems, vol. 30.

Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., Batra, D. (2015):
"Why M heads are better than one: Training a diverse ensemble of deep networks",
http://arxiv.org/abs/1511.06314.

Nado, Z., Band, N., Collier, M., Djolonga, J., Dusenberry, M. W., Farquhar, S., Feng, Q., Filos, A., Havasi, M., Jenatton, R., Jerfel, G., Liu, J., Mariet, Z., Nixon, J., Padhy, S., Ren, J., Rudner, T. G. J., Sbahi, F., Wen, Y., Wenzel, F., Murphy, K., Sculley, D., Lakshminarayanan, B., Snoek, J., Gal, Y., Tran, D. (2022):
"Uncertainty Baselines: Benchmarks for Uncertainty Robustness in Deep Learning"
https://arxiv.org/abs/2106.04015.

Naeini, M. P., Cooper, G. F., Hauskrecht, M. (2015):
"Obtaining well calibrated probabilities using bayesian binning"
Proceedings of the AAAI conference on artificial intelligence, vol. 29.

Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., Dustin, T. (2019):
"Measuring Calibration in Deep Learning",
CVPR Workshops, vol. 2.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014):
"Dropout: A Simple Way to Prevent Neural Networks from Overfitting"
Journal of Machine Learning Research, vol. 15, pp. 1929-1958.

Wen, Y., Tran, D. Ba, J. (2020):
"BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning",
International Conference on Learning Representations.