EXPLAINABLE CLASSIFICATION OF X-RAY MEDICAL IMAGES USING BAYESIAN DEEP LEARNING

Anders David Lægdsgaard Lassen (s210525), Alessandro Contini (s210197),
Mads Birch Sørensen (s195552), Grigor Spalj (s211094)

Technical University of Denmark
Department of Applied Mathematics and Computer Science

ABSTRACT

Deep Neural Networks (DNNs) have often been found to be poorly calibrated due to overconfident predictions [1]. At the same time the increasing complexity of modern DNNs have led to these models being considered black boxes. For that reason, various explanation methods have been proposed to uncover what features influence the predictions of DNNs [2, 3]. Bayesian Deep Neural Networks (BNNs) infer the entire posterior distribution of the weights, meaning that uncertainty about predictions is inherent [4]. In this paper we investigate how a Bayesian approach can improve the calibration and explainability of modern DNNs. We implemented and trained a Convolutional Neural Network (CNN) on the MURA dataset [5] to perform a classification task and found that the Bayesian framework resulted in a significant reduction in calibration error and improved the interpretability of the implemented visual explanation methods. The code has been made available to the public.[1]

1. INTRODUCTION

DNNs have been used with great success within a wide range of fields. However, DNNs are known to suffer from poor calibration as a result of overconfident predictions [1]. At the same time DNN architectures are becoming increasingly complex making it difficult to explain what features the model is basing its predictions on [4]. Together, these issues contribute to DNNs being considered black box models. The black box nature of the DNNs make them less applicable in fields, such as the medical field, where poorly calibrated predictions can be fatal. To investigate the predictions of the DNNs, various visual explanation methods have been developed [2, 3]. However, most of the explanation methods are designed for maximum-a-posteriori (MAP) models. Adopting a Bayesian approach, the entire posterior distribution of the weights is inferred through Laplace approximation and deep ensembles (Eq.1), resulting in inherent uncertainty quantification. Hence, BNNs provide additional information about the uncertainty of predictions, reducing the black box nature of DNNs.

In this paper we investigate if and how a Bayesian approach can improve the calibration visual explanation methods for CNNs.

The posterior distribution of the weights is given by:

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})}. \tag{1}$$

Where θ and $\mathcal{D}$ denote the set of model weights and the dataset respectively. In many cases the normalising constant, the marginal likelihood $p(\mathcal{D})$, is intractable. As a result, we employ 2 approximation methods: the Laplace approximation and the ensemble method.

Our aim is to use explanation methods to highlight areas of interest which contribute most to the decision making of the CNN. This way the expert would not require prior technical knowledge nor have to understand what happens within the model, exposing the black-box nature of the neural network.

2. METHODS

In this section we will introduce the dataset as well as the methods used for approximation of the posteriors and the methods used for explanations.

2.1. Dataset and Model

In this paper we used the MURA dataset [5], a large dataset of musculoskeletal radiographs containing 40,561 images of 7 different bone types. We focused on the application of bone type classification. We split the data into training, validation, and test sets, as done in the original paper. [5]

We implemented and trained two different models; a small CNN with 6 convolutional layers and 4 linear layers, with the last layer having 903 parameters, and a large CNN

[1] `https://github.com/MadsBirch/Bayesian_Explainable_AI`

with 10 convolutional layers and 4 linear layers, with the last layer having 3591 parameters.

2.2. Laplace Approximation

Laplace approximation (LA) is one of the simplest approximation methods for estimating the Bayesian posterior of the weights. LA approximates the posterior with a Gaussian distribution centered at the MAP estimate, with covariance matrix equal to the negative inverse of the local curvature around the MAP estimate:

$$p(\theta \mid \mathcal{D}) \approx \mathcal{N}\left(\theta; \theta_{\mathrm{MAP}}, \Sigma\right), \tag{2}$$

$$\textit{where } \Sigma := -\left(\nabla_\theta^2 \mathcal{L}(\mathcal{D}; \theta)\big|_{\theta_{\mathrm{MAP}}}\right)^{-1}. \tag{3}$$

where $\nabla_\theta^2 \mathcal{L}(\mathcal{D}; \theta)$ denotes the local curvature of the loss function with respect to the parameters θ. We will apply LA post-hoc to approximate the posterior of the weights using the MAP estimate as the mean of Gaussian. Hence, we only need to estimate the Hessian containing the local curvature at the MAP estimate.

In this paper we employ the package developed by Daxberger et. al. 2021 [6]. Since approximating the Hessian is computationally expensive, the authors have implemented different Hessian structures that can be imposed, to reduce computational cost. Choosing a specific Hessian structure results in a trade-off between computational cost and quality of the approximation.

2.3. Ensembles

Deep ensembles approximate the Bayesian posterior of the weights in equation (1) by randomly initializing and training the same model architecture several times [7]. Each model (set of parameters) can then be treated as a sample from the approximate posterior of the weights.

$$\theta_i \sim p(\theta \mid \mathcal{D}). \tag{4}$$

where θ_i is a set of sampled parameters values and D is the dataset. Despite the simplicity of the approximation method, ensembles have proven to perform well in practice [7]. However, for deep ensembles to perform well, functional diversity is important [7]. Functional diversity refers to the fact that the models are converging to different local optima and thereby learning different representations of the data. If models in our ensemble are learning the same representation ($\theta_i \approx \theta_j$), then the models will be redundant in the model averaging.

We initialised and trained our models 15 times, obtaining an ensemble of 15 models. Ensemble predictions were then obtained by taking the mean of the predicted class probabilities across all models.

2.4. Performance Metrics

In this paper we evaluate performance of the approximation methods by considering the F1-score of the predictions because we have imbalanced classes and the expected calibration error (ECE). The ECE measures the difference between classification accuracy and prediction confidences:

$$ECE = \sum_{i=1}^{K} \frac{|B_i|}{N} \left|\mathrm{acc}_i - \mathrm{conf}_i\right|. \tag{5}$$

where K is the number of bins, B_i is the number of predictions in bin i, N is the number of data points, acc_i is the accuracy of bin i and $conf_i$ is the confidence of bin i. Since DNNs are often overconfident in their predictions, we found ECE to be a relevant performance measure; allowing us to investigate how Bayesian uncertainty quantification improves model calibration.

2.5. Gradient explanations

Before defining the gradient explanations, we introduce the notion of a relevance function [4]. It takes an output function $f_\theta : R^d \mapsto R^k$ and maps it to the following:

$$R_\theta(x) = \mathcal{T}_{x,\theta}[f_\theta](x) \tag{6}$$

for some $\mathcal{T}_{x,\theta}$.
We can see the above function depends on the function f_θ but is also directly dependent on the inputs x and weights θ. Gradient explanation is the most basic explanation method and it makes use of the gradient of the relevance function R_θ with respect to the input x.

2.6. Class Activation Maps

As a more advanced visual explanation of the predictions we will present Class Activation Maps (CAMs) for the CNNs. A CAM for a particular category indicates the discriminative image regions used by the CNN to make predictions [8] and are obtained from the parameters of the last convolutional layer in the network. Therefore, changing weights only for the very last layer, the Laplace Approximation network has the same CAMs as the single CNN. The ensembles, on the other hand, will require some aggregation technique to compute the CAMs. To produce the CAMs we sampled 5 CNNs, focusing on 2 classes; HAND and ELBOW. The predictions of the 5 networks are transformed into heatmaps, showing the importance of every area in a blue(least important) to red (most important) color scale, as shown in Figure 2.

2.7. Aggregating explanations

Once the posterior distribution of the weights is estimated accordingly, either by sampling from the Gaussian distribution with LA or using deep ensembles, we obtain new models.

The resulting models can, therefore, be evaluated and new explanations may be extracted given the model and input. Those different explanation methods can be aggregated, which provide an adequate summary of the explanations across the models and may produce better interpretation of the decision making of the models. We will use 3 aggregation methods [4], which are given below:

1. Mean: by taking the average of the explanations, we discover what the average model takes into account during prediction.
2. Intersection: this aggregation method may help us identify the most important region of interest, as we only extract the regions which were considered by every model. This is achieved by calculating the minimum of each pixel across the explanations produced by different models.
3. Union: an implicit method which may be used to identify regions that serve no purpose in decision-making of the model, and is achieved analogously as the intersection except here we calculate the maximum across the models.

Model	Posterior Approx.	F1-score ± SE (%)	ECE ± SE (%)
Large CNN (3591)	MAP	88.4 ± 0.6	74.2 ± 0.8
	LA (Diagonal)	88.4 ± 0.6	13.9 ± 0.2
	LA (Kronecker)	88.4 ± 0.6	13.1 ± 0.3
	LA (Full)	88.4 ± 0.6	13.5 ± 0.3
	Ensemble (2)	90.0 ± 0.8	2.5 ± 1.0
	Ensemble (5)	90.7 ± 0.4	3.1 ± 0.6
	Ensemble (10)	90.7 ± 0.3	4.0 ± 0.5
	Ensemble (15)	90.8 ± 0.2	3.8 ± 0.5
Small CNN (903)	MAP	82.7 ± 0.5	69.2 ± 0.7
	LA (Diagonal)	82.7 ± 0.5	12.2 ± 0.3
	LA (Kronecker)	82.7 ± 0.5	12.1 ± 0.3
	LA (Full)	82.7 ± 0.5	12.7 ± 0.3
	Ensemble (2)	83.9 ± 1.4	3.3 ± 0.8
	Ensemble (5)	85.4 ± 0.8	3.6 ± 1.1
	Ensemble (10)	85.5 ± 0.4	4.2 ± 0.7
	Ensemble (15)	85.5 ± 0.2	4.2 ± 0.2

Table 1: The above results were obtained over 15 independent iterations of: *training model, performing posterior approximation and then evaluating performance on the validation set.* MAP denotes the maximum a posteriori estimate of the parameters.

3. RESULTS

Applying post-hoc Laplace approximation using three different Hessian structures; *diagonal*, *Kronnecker* and *full*, we found that accuracy was unaffected, but the ECE decreased dramatically for all three LA methods (Table 1). We found no significant difference in ECE between the three Hessian structures. We investigated the computational cost of the three different Hessian structures and found that the full Hessian was the fastest.

Using deep ensembles of models as an approximation of the Bayesian posterior, increases prediction F1-score significantly and decreases ECE for both model architectures substantially as seen in Table 1.

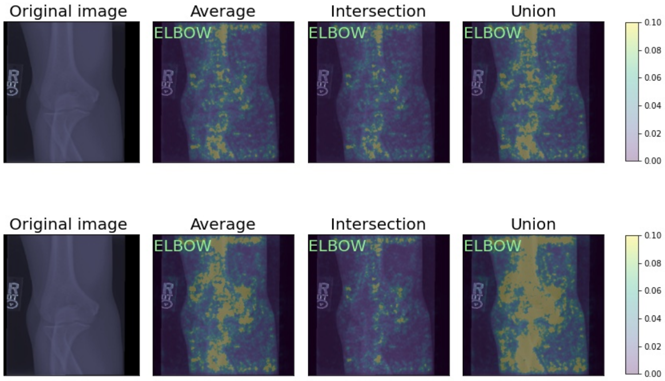


Fig. 1: Gradient explanations of an X-ray image using models whose last layer is sampled with the Laplace approximation (top row) and using models generated with the ensemble method (bottom row). Correctly predicted class is given in the top-left corner of the explanations.

3.1. Gradient explanations

Here we investigate the gradient explanations produced by the models using the LA and the ensemble method. Top row of figure 1 shows the elbow x-ray image together with the aggregated explanations using the Laplace approximation on the last layer of the CNN, sampled 15 times. The Hessian matrix was approximated using a block-diagonal Kronecker structure. We can see there is no significant difference between the aggregations, which can be explained by the fact the models differ in only the last layer.

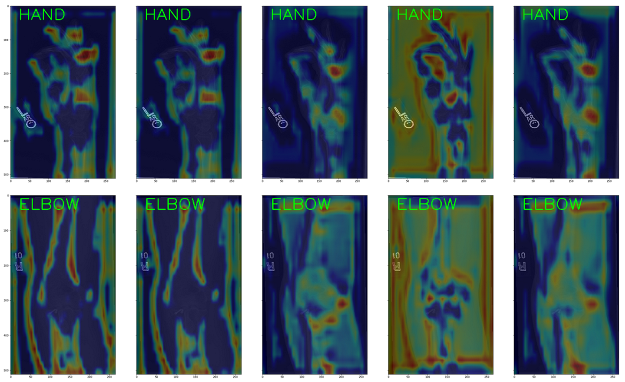
On the other hand, given the 15 models trained for the deep ensemble method, we can see in the bottom row of figure 1 there is a clear distinction between the aggregated explanations, indicating a difference in the regions of interest of each model; some might focus on the bone structure and joints, while another might learn from the shape of the arm part.

3.2. Class Activation Maps

The CAMs for each of the models in the ensemble can be seen in Figure 2. It is apparent that the models in the ensemble have learned different optimal solutions. Fig. 3 shows the the

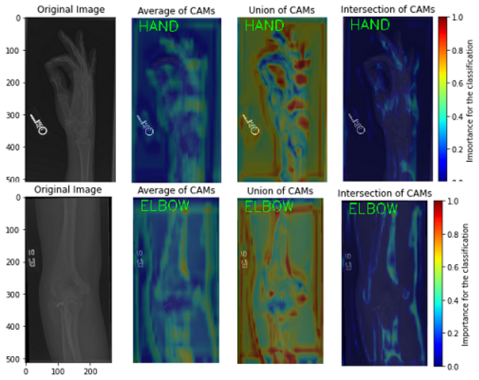
CAMs of the ensemble using the different aggregation methods. We see that the diverse set of optimal solutions learned by the ensemble, results in the union aggregation showing high activation for most areas of the images.

Fig. 2: The CAMs of the individual models composing the ensemble.



3.2.1. *BNN Activation Maps*

Fig. 3: The CAMs of the ensemble.

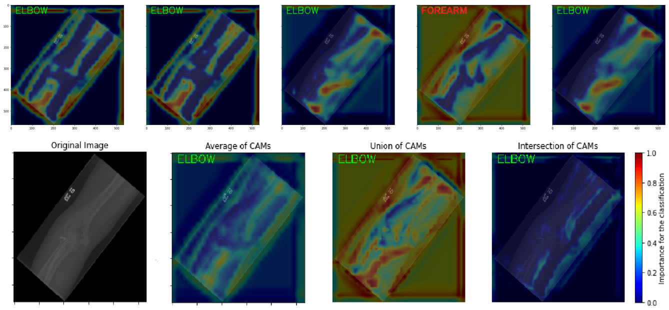


We can clearly see that all the aggregation methods highlight, in terms of classification relevance, the most important parts of the images, in our case the contours of the body parts. In particular, the average aggregation is interesting to us because it illustrates the result of combining the different optimal solutions into a single CAM. Hence, we observe that using the Bayesian framework provided useful information about the uncertainty of the produced CAMs (Fig. 2).

3.2.2. *BNN Rotational Invariancy*

CNNs are not naturally equivariant to some transformations, like rotation of an image. Other mechanisms are necessary for handling these kinds of transformations [9]. The presented BNN is trained with randomly rotated images to handle this problem. The behaviour of the network is verified by passing a 45-degrees rotated elbow image through the aforementioned process.

Fig. 4: Rotated elbow classification.



We can notice (first row of Fig. 4) that the 4th model of the ensemble is making a wrong classification. Still mean prediction computed by the BNN predicts the correct label.

4. CONCLUSIONS

We found that using a Bayesian framework, resulted in a dramatic decrease in calibration error. Additionally, by aggregating the gradient explanations and CAMs over the set of models, we discovered that the models had learned optimal but different solutions. By looking at the diverse set of learned representation, valuable information about the certainty of predictions can be gained.

By investigating what information is most important for the models when making predictions, we gained valuable insight into the certainty about the learned representations, and thereby about the certainty about the predictions of the models. Hence, by adopting a Bayesian framework we successfully reduced the black-box nature of modern CNNs.

However, we found that the methods used for visual explanations were highly sensitive to rotation(Fig. 4), suggesting these visual explanations should be interpreted with caution.

5. REFERENCES

[1] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger, “On calibration of modern neural networks,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 1321–1330.

[2] William Falcon, “4 reasons why companies struggle to adopt deep learning,” 2018.

[3] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Mueller, “How to explain individual classification decisions,” 2009.

[4] Kirill Bykov, Marina M.-C. Höhne, Adelaida Creosteanu, Klaus-Robert Müller, Frederick Klauschen, Shinichi Nakajima, and Marius Kloft, “Explaining bayesian neural networks,” *CoRR*, vol. abs/2108.10346, 2021.

[5] Pranav Rajpurkar, Jeremy Irvin, Aarti Bagul, Daisy Ding, Tony Duan, Hershel Mehta, Brandon Yang, Kaylie Zhu, Dillon Laird, Robyn L. Ball, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng, “Mura: Large dataset for abnormality detection in musculoskeletal radiographs,” 2017.

[6] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig, “Laplace redux-effortless bayesian deep learning,” *Advances in Neural Information Processing Systems*, vol. 34.

[7] Andrew Gordon Wilson, “The case for bayesian deep learning,” *arXiv preprint arXiv:2001.10995*, 2020.

[8] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization.,” *CVPR*, 2016.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016, `http://www.deeplearningbook.org`.