Comparing a deterministic and a Bayesian classification neural network for chest diseases in radiological images

Jonas Nolde and Ruxandra Lasowski

Hochschule Furtwangen University
jonas.nolde@hs-furtwangen.de,
ruxandra.lasowski@hs-furtwangen.de

Abstract. A common mantra for automated decision systems is that a system should know when it doesn't know. Bayesian neural networks are designed to capture uncertainties over the network weights and in theory, they perform better predictions and output uncertainties. To this end, we compare in this paper a deterministic neural network and a Bayesian neural network for the classification of chest diseases in radiological images. We use the ChestX-ray14 data set [1] involving 14 respiratory diseases like pneumonia and atelectasis. We found that the deterministic network similar to CheXNet [2] outperformed the Bayesian version in this task, whereas, employed on the more simplistic MNIST dataset it did not. Our experiments suggest that there is a gap between theory and practical use of BNNs for very deep networks and real clinical data.

Keywords: medicine, radiology, classification, deterministic neural networks, Bayesian neural networks

1 Introduction

In 2016 almost 2.38 million people died from lower respiratory infections worldwide. This was the sixth leading cause of mortality for all ages that year [3]. For the last three decades, pneumonia was the most common cause of death for children under 5 years [4]. While the treatment of a diagnosed pneumonia patient can be done efficiently with low-cost, low-tech medication and care [5], the detection of lower respiratory infections leaves room for improvement. The diagnosis of such diseases with the help of chest X-rays is very effective and currently the best available method [6]. Unfortunately, this task is challenging and requires expert radiologists which are rare in impoverished regions.

Past work about the detection and classification of lung diseases on chest X-rays with convolutional neural networks (CNNs) could achieve promising results that match or exceed the performance of radiologists [2]. These deterministic neural networks, albeit being right most of the time, won't tell you how certain they are about their decisions. In critical applications, where the cost of error is high, an indication of confidence can be extremely valuable, especially in uncertain edge cases.

While deterministic neural networks contain a specific set of weights, Bayesian neural networks (BNNs) assign probabilities to all possible sets of weights, allowing for uncertainty quantification. In this work, we will assess the applicability of Bayesian deep learning in the field of medical diagnosis. We, therefore, implement a version of the deterministic neural network CheXNet [2] and a Bayesian version of CheXNet using recent advances in Bayesian deep learning. We then evaluate and compare their performance on the *ChestX-ray14* data set [1]. Furthermore, we efficiently measure aleatoric and epistemic uncertainties in the Bayesian model's predictions.

2 Related Work

2.1 Bayes' Theorem

Bayesian deep learning utilizes *Bayes' Theorem* to calculate conditional probabilities. In the context of deep learning, Bayes' Theorem can be rewritten as

$$p(w|\mathcal{D}) \;=\; \frac{p(\mathcal{D}|w)\,p(w)}{p(\mathcal{D})}, \tag{1}$$

with a neural network's parameters w (the *hypothesis*) and data $\mathcal{D}$ (the *evidence*). Bayesian deep learning aims to calculate the *posterior* distribution $p(w|\mathcal{D})$, which "captures the set of plausible model parameters, given the data" [7]. This is done by multiplying the *likelihood* $p(\mathcal{D}|w)$ of data $\mathcal{D}$ occurring given parameters w with the *prior* distribution $p(w)$ and normalizing it by the data distribution $p(\mathcal{D})$.

We can implement a Bayesian neural network by replacing a "deterministic network's weight parameters with distributions over these parameters, and instead of optimising the network weights directly we average over all possible weights (referred to as *marginalisation*)" [7]. With *Bayesian inference* we can calculate the posterior distribution $p(w|\mathcal{D})$ and predict probabilities y^* given new data x^*:

$$p(y^*|x^*,\mathcal{D}) \;=\; \int p(y^*|x^*,w)\;p(w|\mathcal{D})\,dw\;. \tag{2}$$

2.2 Variational Inference

Hinton and Van Camp initially proposed *variational inference* for neural networks in 1993 [8] as an alternative to methods involving expensive Monte Carlo sampling. In 2011, Graves [9] published an improved approach, suitable for more complex neural networks. Variational inference solves the intractability of the integral over the true posterior distribution p in eq. 2 by integrating over a simplified posterior distribution q with variational parameters θ instead. This variational posterior q_θ is an approximation of the true posterior p but is easier to sample from. In most cases, a Gaussian distribution $\mathcal{N}(\mu,\sigma^2)$ with parameters $\theta=(\mu,\sigma^2)$ for the mean and the variance respectively is used. Graves' approach optimizes the posterior approximation $q_\theta(w|\mathcal{D}) \approx p(w|\mathcal{D})$ by minimizing the variational free energy $\mathcal{F}$, also referred to as the negative variational lower bound or negative evidence lower bound (ELBO). For deep learning, $\mathcal{F}$ can be reinterpreted as *minimum description length cost function* [9]:

$$\mathcal{F}(\mathcal{D},\theta) \;=\; KL[q_\theta(w)||p(w)] - \mathbb{E}_{q_\theta(w)}[log\;p(D|w)], \tag{3}$$

where $KL[q_\theta(w)||p(w)]$ is the *Kullback-Leibler (KL) divergence* between both distributions. It consists of a data-dependent part (the *likelihood cost* or *error loss*) and a prior-dependent part (the *complexity loss*). The function embodies a trade-off between satisfying the complexity of the data $\mathcal{D}$ and the simplicity of the prior $p(w)$ [10].

Graves' approach for a tractable approximation of the Bayesian neural network's posterior distribution was the cornerstone for more efficient and stable estimation methods like Stochastic Gradient Variational Bayes (SGVB) [11], the local reparameterization trick [12] and the flipout estimator [13].

2.3 Uncertainty Estimation in Bayesian Deep Learning

Kendall and Gal [7] describe the two types of uncertainty in the context of Bayesian deep learning as follows:

Epistemic uncertainty is often referred to as *model uncertainty*, as it "captures our ignorance about which model generated our collected data" [7].

Aleatoric uncertainty, on the other hand, "captures noise inherent in the observations" [7] and thus is often referred to as *data uncertainty*. This type of uncertainty can further be categorized into *homoscedastic* uncertainty and *heteroscedastic* uncertainty. While homoscedastic uncertainty stays constant for different inputs, heteroscedastic uncertainty varies from input to input, with some potentially having more noisy outputs than others. They note that "heteroscedastic uncertainty is especially important for computer vision applications" [7].

The paper concludes, that measuring both types of uncertainty is crucial for the safety and reliability of models. Aleatoric uncertainty is important for "large data situations, where epistemic uncertainty is explained away" and "real-time applications, because we can form aleatoric models without expensive Monte Carlo samples" [7]. Epistemic uncertainty is important for "safety-critical applications, because epistemic uncertainty is required to understand examples which are different from training data" and "small datasets where the training data is sparse" [7].

Kendall and Gal [7] proposed a method to estimate both the aleatoric and the epistemic uncertainty, which was later refined for classification by Kwon et al. [14]:

$$\underbrace{\frac{1}{T}\sum_{t=1}^{T}\operatorname{diag}(\hat{p}_t) - \hat{p}_t\,\hat{p}_t^T}_{\text{aleatoric}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}(\hat{p}_t - \overline{p}_t)(\hat{p}_t - \overline{p}_t)^T}_{\text{epistemic}}, \tag{4}$$

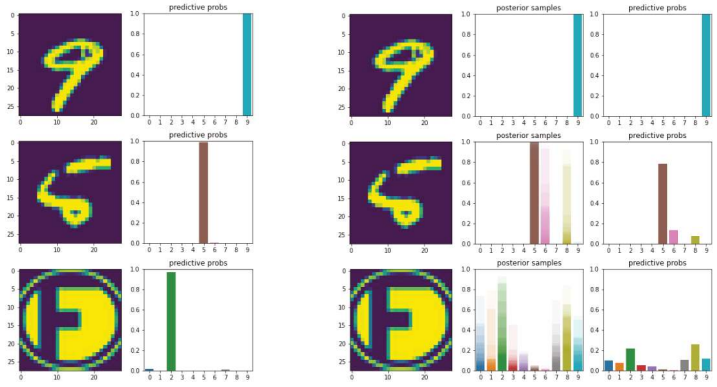
with the predicted probability vector $\hat{p}_t = p(\hat{w}_t) = \text{Softmax}\{f^{\hat{w}_t}(x^*)\}$, that is sampled T times, the diagonal matrix $\operatorname{diag}(\hat{p}_t)$ with elements of vector $\hat{p}_t$, and the mean predicted probability $\overline{p} = \sum_{t=1}^{T} \hat{p}_t / T$. We later use the formula of 4 in our experiment to measure our model's uncertainties.

3 Methods

3.1 First Tests with TensorFlow Probability and MNIST

To gain our first practical experience in implementing a Bayesian convolutional neural network for disease classification on X-rays with TensorFlow 2.0 and its probabilistic programming library TensorFlow Probability, we first looked at a simpler image classification task. We implemented the small LeNet-5 network [15]. It consists of only 7 layers (3 convolutional layers, 2 subsampling layers, 2 fully connected layers) making it a relatively simple and small network in today's time. The original LeNet-5 architecture is a deterministic neural network and does not use probabilistic methods. To make it Bayesian we replaced the deterministic convolutional and fully connected layers from TensorFlow with the probabilistic layers from TensorFlow Probability. The model's task is to classify the images of the *MNIST* data set [15] and tell which digit is depicted. The data set contains 28×28 pixel small, grayscale images of a handwritten digit (0-9), normalized in size and centered in the image and is split into 60,000 training samples and 10,000 test samples. We trained the model in mini-batches with 128 normalized images each, where the Adam optimizer [16] minimizes the categorical cross-entropy loss.

Results and Conclusion After training for 75 epochs, the Bayesian model achieved a validation accuracy of 0.984, which was good enough for us to stop the training. We then performed Monte Carlo sampling by asking the trained model to predict the labels of the same unseen images 50 times. The resulting outputs were different at each prediction as figure 1(b) shows. For comparison, we trained a non-Bayesian, deterministic version of the network. 10 epochs of training already yielded a training accuracy of 0.997. Figure 1(a) shows the resulting test prediction plots. Note that a deterministic model always outputs the same values for the same input, requiring only one prediction per sample at inference. To show how the models perform on unusual data, we tested with "fake" MNIST images from the *notMNIST* database, which contains images that look similar to those in the MNIST database but show letters instead of numbers. The resulting prediction plots can be seen in the last samples of figure 1.



(a) Deterministic model (b) Bayesian model

Fig. 1: Predictions of the deterministic and the Bayesian model on the MNIST data set.

Concluding, the first thing to note is that training the Bayesian neural network takes considerably longer than training the deterministic one. Figure 1 shows that both models correctly classified the first sample. However, the Bayesian model predicted wrong classes several times during the Monte Carlo inference on the second sample. Looking at the input image, we can argue that the "5" looks like a "6" or an "8" to some degree. Thus, having the Bayesian model predict those digits a few times is a justifiable and possibly desirable sign of uncertainty. The third sample depicts the letter "F" in a circle and is also inverted. Hence, the sample neither belongs to a class that can be predicted by the model nor did the model see comparable images during training. While the deterministic model was sure it saw the digit "2" (figure 1(a)), the Bayesian model was highly uncertain in its decision (figure 1(b)). The plot shows that during Monte Carlo sampling the model predicted different classes every time, causing the average prediction to have a low, almost similar probability for every class. This can be interpreted as an indication of high uncertainty and the model saying "I don't know". In critical applications like medical diagnosis, this behavior is highly desirable.

3.2 CheXNet and Bayesian CheXNet

The main goal of this work is to show the advantages of Bayesian deep learning over traditional, deterministic deep learning for medical image classification. We compare performances of the deterministic CheXNet [2] and our Bayesian neural network with a similar architecture.

CheXNet achieves state-of-the-art results on all 14 diseases of the publicly available *ChestX-ray14* data set [1]. The data consists of 112,120 single grayscale image files and CSV files with metadata like the images' disease labels and bounding boxes indicating the location of the disease. We pre-processed the images by down-scaling them to 224×224 pixels and normalizing the 8-bit pixel values (0 - 255) to float values between −1 and 1. Finally, we split the data into train, validation, and test set containing 98,656 (93.5%), 6,336 (6%), and 432 (0.5%) data points respectively.

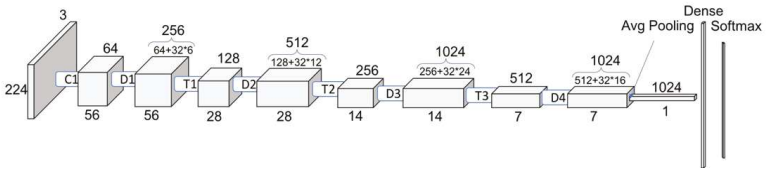


Fig. 2: DenseNet121 architecture with Dense Blocks D and Transition Blocks T; Source: [17]

To replicate CheXNets model architecture, we used a copy of the *DenseNet121* model (depicted in figure 2) coming with TensorFlow's *Keras* API. We changed the input shape of the model to match the monochromatic 224×224 image matrices. Furthermore, we changed the output layer to return 14 values and used the *sigmoid* non-linearity activation function instead of *softmax*, as we want the model to predict independent continuous values to indicate each disease. For our Bayesian version of CheXNet, we replace the deterministic convolutional and fully connected layers with probabilistic layers from the TensorFlow Probability library.

Training Results We trained both models with mini-batches by minimizing the binary cross-entropy loss with the Adam optimizer. After 70 epochs, the deterministic model achieved a training AUC of 0.9006 and a validation AUC of 0.8582. We outperform the original CheXNet with an average per-class AUC score of 0.8414 by a small margin. Our Bayesian model failed to achieve good performance just by introducing the probabilistic layers. We stopped the training after 17 epochs as the validation loss started to increase while the AUC decreased to 0.5 which is the random baseline.

3.3 Model Improvement Approaches

As our first results show, we couldn't achieve good model performance in this task simply by using Bayesian layers. Although we could get additional benefit from implementing uncertainty measures, a bad performing model isn't practically useful in any real-world application, let alone in disease detection. In the following, we will discuss and test several approaches we took to increase the model's performance.

Choosing Hyperparameters In our first approach, we searched for better hyperparameters by training the neural network with different handpicked sets of *learning rates*, *mini-batch sizes*, and *optimization algorithms*. The model that performed best was trained with the Adagrad optimizer, a learning rate of 0.1, and a mini-batch size of 64. An automated search for parameters with *Hyperopt* [18] led to similar results.

Dealing with Imbalanced Data As our data set is extremely imbalanced (with significantly more negatives than positives), we introduce a *weighted loss function* that penalizes misclassified positive samples more. This way we can nudge the model towards looking at positive training samples more carefully. This approach, which was also proposed in the CheXNet paper, significantly improved our model's learning and performance on the F1-score and the AUC value.

Initialization with Pre-trained Weights The deterministic CheXNet is initialized with weight parameters pre-trained on the *ImageNet* data set. In an attempt to make use of parameter transfer learning in our Bayesian model, we initialize the model's priors with normal distributions with variance 1, mean-shifted towards the single-point parameter values from the pre-trained weights. After training for a while, we concluded that the initialization with weights pre-trained on *ImageNet*, neither improved model performance nor sped up the training.

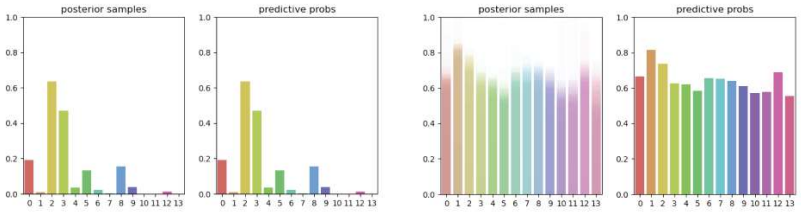
4 Results and Conclusion

Test set	AUC	F1-score	F2-score	Epistemic	Aleatoric
Deterministic					
Full	0.8339	0.1444	0.1019	-	-
1	0.7552	0.2858	0.2000	-	-
2	0.6940	0.2858	0.2000	-	-
3	0.9502	0.0000	0.0000	-	-
4	0.8523	0.3333	0.2778	-	-
5	0.9091	0.4000	0.2941	-	-
6	0.7614	0.0000	0.0000	-	-
Bayesian					
Full	0.6579	0.1298	0.2551	-	-
1	0.5378	0.1176	0.1923	0.0111	0.2098
2	0.4544	0.1176	0.1923	0.0143	0.2088
3	0.4851	0.0571	0.1135	0.0033	0.2149
4	0.6098	0.0851	0.1695	0.0076	0.2129
5	0.5739	0.0976	0.1888	0.0109	0.2080
6	0.6686	0.1499	0.2884	0.0037	0.2166

Table 1: Test results of the deterministic and the Bayesian model on each test set.

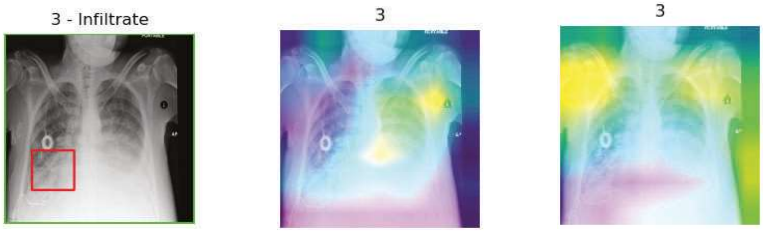
We assessed the deterministic and the Bayesian models' performance on the test set and additional samples with Gaussian noise. Our deterministic model achieved an AUC

of 0.8339, similar to the 0.8414 stated in the original CheXNet paper. This verifies that our implementation of the model and the rest of our deep learning pipeline work as expected. The Bayesian model achieved a lower AUC of 0.6579, as well as lower F1-, and F2-scores. We measured mean *epistemic* uncertainties ("model uncertainty" [7]) of $0,0085$ and mean *aleatoric* uncertainties ("data uncertainty" [7]) of $0,2118$.



(a) Deterministic model (b) Bayesian model

Fig. 3: The models' predictions for a test sample.



(a) Ground Truth (b) Deterministic model (c) Bayesian model

Fig. 4: The models' CAMs for class 3 of a test sample and the actual location of the disease.

Figure 3 shows an example of the models' predicted probabilities for a test sample with true labels 2 and 3. The deterministic model predicted classes 2 and 3 with the threshold of 0.5, while the Bayesian model's averaged predictions of the 50 predictions sampled show all classes to be true. The generated class activation maps (CAMs) of figure 4 show that both models didn't look at the right location for their prediction of class 3.

We interpret the much higher aleatoric uncertainty as a result of the nature of radiological images, which can also contain pacemakers and/or other patient-specific aids and the low resolution of the images that were fed into the network. The epistemic uncertainty suggests that the prior for the model should be adjusted. The sampled posterior probabilities range between 0.6 and 0.8 for each class on most of the test samples, which could imply that the model's weight distributions were initialized with a too high variance that couldn't be reduced during training. So far, state-of-the-art results with

Bayesian neural networks were achieved on simplistic and carefully curated data sets like MNIST and CIFAR-10 moderate deep networks [19]. Our experiments with radiological images and very deep networks didn't achieve state-of-the-art results. This suggests that the complexity of the data, the model size, and/or the initialized variance are the most important factors that can be further analyzed.

References

1. Summers, R.: Nih chest x-ray dataset of 14 common thorax disease categories. https://nihcc.app.box.com/v/ChestXray-NIHCC/file/220660789610 (2017) Accessed: 2020-07-27.
2. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D.Y., Bagul, A., Langlotz, C., Shpanskaya, K.S., Lungren, M.P., Ng, A.Y.: Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. CoRR **abs/1711.05225** (2017)
3. Troeger, C., Blacker, B., Khalil, I.A., Rao, P.C., Cao, J., Zimsen, S.R., Albertson, S.B., Deshpande, A., Farag, T., Abebe, Z., et al.: Estimates of the global, regional, and national morbidity, mortality, and aetiologies of lower respiratory infections in 195 countries, 1990–2016: a systematic analysis for the global burden of disease study 2016. The Lancet infectious diseases **18**(11) (2018) 1191–1210
4. Max Roser, H.R., Dadonaite, B.: Child and infant mortality. Our World in Data (2013)
5. World Health Organization (WHO): Pneumonia. https://www.who.int/news-room/fact-sheets/detail/pneumonia (2019) Accessed: 2020-07-29.
6. Kesselman, A., Soroosh, G., Mollura, D.J., Abbey-Mensah, G., Borgstede, J., Bulas, D., Carberry, G., Canter, D., Ebrahim, F., Escalon, J., et al.: 2015 rad-aid conference on international radiology for developing countries: the evolving global radiology landscape. Journal of the American College of Radiology **13**(9) (2016) 1139–1144
7. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in neural information processing systems. (2017) 5574–5584
8. Hinton, G.E., Van Camp, D.: Keeping the neural networks simple by minimizing the description length of the weights. In: Proceedings of the sixth annual conference on Computational learning theory. (1993) 5–13
9. Graves, A.: Practical variational inference for neural networks. In: Advances in neural information processing systems. (2011) 2348–2356
10. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural networks (2015)
11. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
12. Kingma, D.P., Salimans, T., Welling, M.: Variational dropout and the local reparameterization trick. In: Advances in neural information processing systems. (2015) 2575–2583
13. Wen, Y., Vicol, P., Ba, J., Tran, D., Grosse, R.: Flipout: Efficient pseudo-independent weight perturbations on mini-batches. arXiv preprint arXiv:1803.04386 (2018)
14. Kwon, Y., Won, J.H., Kim, B.J., Paik, M.C.: Uncertainty quantification using bayesian neural networks in classification: Application to ischemic stroke lesion segmentation. (2018)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. Ruiz, P.: Understanding and visualizing densenets. https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a (2018) Accessed: 2020-08-18.
18. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. The Journal of Machine Learning Research **13**(1) (2012) 281–305
19. Shridhar, K., Laumann, F., Liwicki, M.: A comprehensive guide to bayesian convolutional neural network with variational inference. CoRR **abs/1901.02731** (2019)