Machine Learning (CS282) Final Project: Uncertainty Estimation Using a Single Deep Deterministic Neural Network

Zhitong Gao [*1] **Ziyao Zeng** [*1]

Abstract

DUQ (Van Amersfoort et al., 2020) is a recently proposed deterministic uncertainty estimation method for deep neural networks, where an adapted RBF network is trained with a two-sided gradient penalty, and the feature distance between data is used to predict uncertainty. The paper claims to outperform deep ensemble (Lakshminarayanan et al., 2016) in the task of out of distribution(OoD) detection on two dataset pairs. In this report, we aim to provide a critical evaluation of DUQ through methodological analysis and experimental reproduction. Along with the evaluation, we identify potential research areas to explore next.

1. Introduction

With more and more machine learning algorithms participating in the human decision-making process, it is vital to estimate the reliability of model predictions, especially in high-risk areas. Uncertainty estimation has seen increased attention in recent years and plays an important role in AI settings such as guiding exploration in Reinforcement Learning (Osband et al., 2016), and data selection in Active Learning(Houlsby et al., 2011).

Recent advances have been made in deep learning for predictive uncertainty estimation. One such approach is based on Bayesian Neural Networks (BNNs), where model parameters are described by a distribution rather than by a deterministic value. One of the most popular methods is Monte Carlo (MC) dropout (Gal & Ghahramani, 2016), which avoids the difficulty of posterior inference approximation and adopts dropout sampling in each layer of the network. Uncertainty is estimated by conducting several forward passes at test time. Some non-Bayesian approaches have also been proposed. Deep Ensembles (Lakshminarayanan et al., 2016), as one of these methods, is reported to outperform MC dropout. It utilizes the training of multiple networks with different initializations and different data orders and uses the entropy of predictions to obtain uncertainty. Though effective, its performance comes at the cost of high computational cost.

To address the problem, Van et al. propose a non-bayesian method called "Deterministic Uncertainty Quantification (DUQ)" (Van Amersfoort et al., 2020), which aims to improve computational efficiency by estimating uncertainty within only one forward pass. Specifically, the paper proposes an uncertainty estimation method by measuring samples' distances with closest class centroids in feature space. Training is conducted as minimizing distance loss. The centroids updating and loss function are designed to aid stable optimization. To encourage sensitivity while maintaining generalization ability, a gradient penalty is used. Experiments conducted on one small dataset "Two Moons" and two out of distribution(OoD) dataset pairs: FashionMNIST vs MNIST, CIFAR-10 vs SVHN have validated the method's feasibility.

The scope of this report is to provide a critical evaluation of the "Deterministic Uncertainty Quantification (DUQ)" method. On the one hand, we are appreciated by the idea of the deterministic way of estimating predictive uncertainty, which is quite different from the previous work. The simplicity of the method also makes it easy to follow, re-implementation, and make changes. Some of the ideas explained in the paper are also enlightening, such as the gradient penalty used to balance the sensitivity and generalization ability, the distance-aware training strategy, etc. However, the methodology itself could exist some limitations which are not explained in the original paper, for example, the gradient penalty may fail with the residual block design, which is quite common in neural network structure. Besides the methodological discussion, we further conduct experiments mentioned in the paper, to check the feasibility of the methods in the empirical view. Lastly, we propose a potential improvement direction of separating two kinds of uncertainties. Our tiny experiments show the potential utility of this proposal.

[*]Equal contribution [1]School of Information Science and Technology, ShanghaiTech University. Correspondence to: Zhitong Gao <gaozht@shanghaitech.edu.cn>, Ziyao Zeng <zengzy@shanghaitech.edu.cn>.

Machine Learning (CS282) Final Project, ShanghaiTech University, 2021 Spring

2. DUQ Method

The design of the DUQ method can be mainly divided into two parts. The first part tries to stabilize the training of RBF networks by using a binary cross entropy loss and updating centroids with momentum. The second part encourages model sensitivity while enforcing output smoothness by incorporating a two-sided gradient penalty. Following, we first introduce the proposed methods with Section 2.1 focusing on tricks for RBF network stabilization and Section 2.2 describing the gradient penalty design. Then in Section 2.3, we review the method with three aspects: contribution, strengths, and weaknesses. A depiction of the architecture of DUQ is shown in Figure 1.

2.1. RBF network Stabilization

In DUQ, a deep network without softmax layer is used to map original data into feature space. As in normal RBF networks, an RBF kernel is then applied to measure feature distance with class centroids. Formaly, we denote the extracted feature for input $x \in \mathbb{R}^m$ as $f_\theta(\mathbf{x}) \in \mathbb{R}^d$, where m is the input dimension, d is the feature dimension and θ is the parameter of the deep network f. Centroid for class c is denoted as K_c. The distance between feature vector $f_\theta(\mathbf{x})$ and class centroid K_c is:

$$K_c\left(f_\theta(\mathbf{x}), \mathbf{e}_c\right) = \exp\left[-\frac{\frac{1}{n}\left\|\mathbf{W}_c f_\theta(\mathbf{x}) - \mathbf{e}_c\right\|_2^2}{2\sigma^2}\right] \quad (1)$$

where W_c is a learnable class-wise weight matrix, of size n (centroid size) by d (feature extractor output size). σ is a hyper parameter called the length scale.

During training, centroids are firstly fixed, and network paramer θ and weight vector $\mathcal{W} = \{\mathbf{W}_1, \cdots, \mathbf{W}_c\}$ are updated using a binary cross entropy(BCE) loss, defined as

$$L(\mathbf{x}, \mathbf{y}) = -\sum_c y_c \log\left(K_c\right) + \left(1 - y_c\right) \log\left(1 - K_c\right) \quad (2)$$

where $\{\mathbf{x},\mathbf{y}\}$ is a data point in the training data set $\{X,Y\}$, with $\mathbf{y}$ being one-hot encoding. Intuitively, the BCE loss enforces each data to be closed to its given class centroid, and far away from other class centroids.

After that, parameters are held constant, and centroids $\mathbf{E}$ are updated using an exponential moving average of the feature vectors of data points belonging to that class. Specifically,

$$\begin{aligned} n_{c,t} &= \gamma * n_{c,t-1} + (1-\gamma) * n_{c,t} \\ \mathbf{m}_{c,t} &= \gamma * \mathbf{m}_{c,t-1} + (1-\gamma) \sum_i \mathbf{W}_c f_\theta\left(\mathbf{x}_{c,t,i}\right) \\ \mathbf{e}_{c,t} &= \frac{\mathbf{m}_{c,t}}{n_{c,t}} \end{aligned} \quad (3)$$

where $n_{c,t}$ is the number of data points assigned to class c in minibatch t. $\mathbf{x}_{c,t,i}$ is element i of a minibatch at time t, with

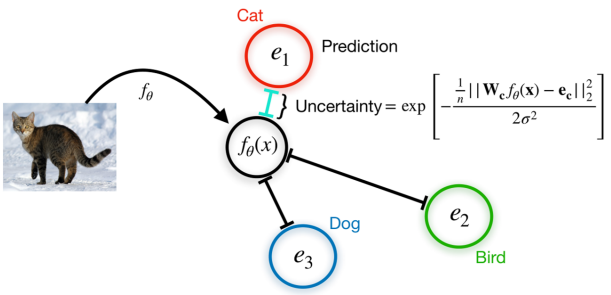


Figure 1. The structure of DUQ. The input image is first mapped to the feature space and then predicted as the label corresponding to the closest centroid. This closest distance represents the uncertainty of the prediction.

class c. γ is the momentum coefficient, which is usually set between $[0.99, 0.999]$. The high momentum stablizes the optimization but makes it hard to converge, for which, an L2 normalization term of θ is added.

In inference, the prediction $\mathbf{y}_{pred}$ is made by taking the class c with closest distance:

$$\mathbf{y}_{pred} = \arg\max_c K_c\left(f_\theta(\mathbf{x}), \mathbf{e}_c\right) \quad (4)$$

And the corresponding predictive uncertainty $U(\mathbf{x})$ is defined as the distance to the closest centroid:

$$U(\mathbf{x}) = \max_c K_c\left(f_\theta(\mathbf{x}), \mathbf{e}_c\right) \quad (5)$$

2.2. Gradient Penalty

As described above, the proposed model is trained to minimize the distance between data feature and corresponding class centroid, while maximizing others. This optimization process forces the model to push every data representation near to centroids, making it possible to map out-of-distribution data to in-distribution space, which is called *feature collapse* in the investigated paper. To avoid feature collapse means to make the model sensitive to input space. In other words, a change in input space can be noticed in feature space.

Motivated by this, the authors propose to add a two-sided gradient penalty in the training process,

$$\lambda \cdot \left[\left\|\nabla_x \sum_c K_c\right\|_2^2 - 1\right]^2 \quad (6)$$

where $\sum_c K_c$ is limited to the targeted Lipschitz constant 1. The author explains the intuition behind this two-sided gradient penalty. For one side, the gradient can not be too

large (larger than 1) to enforce smoothness. On the other side, the gradient is not expected to be too small (less than 1) to encourage sensitivity to input changes.

2.3. Methodology Review

2.3.1. Summary and Contributions

This paper proposes a deterministic method for predictive uncertainty estimation. Based on RBF networks, the authors use BCE loss and update centroids with momentum to stabilize the training. A two-sided gradient penalty is incorporated to avoid feature collapse.

2.3.2. Strengths

- Novel idea of adapting RBF networks to estimate predictive uncertainty in a deterministic way. Previous methods mostly use non-deterministic methods, represented by MC dropout and Deep Ensembles, and are much more time-consuming.

- Novel design of using gradient penalty to balance sensitivity and generalization ability. These two seem to be a tradeoff. If a model is very sensitive to input, in other words, a slight difference in input space can be detected in feature space. It is good for OoD detection but bad for generalization ability. Also, the classification task itself aims to map different inputs to a single class, which may enforce feature collapse.

- Good design of applying momentum strategy in centroids updating. The momentum term maintains the centroid's past information, stabilizing the optimization.

- Good discussion of the weakness of softmax-based methods. Normally, the discriminate models focus on decision boundaries. If an OoD data is far away from the decision boundaries, its uncertainty may be very low under the evaluation of prediction entropy, since the model only makes different decisions near the decision boundary.

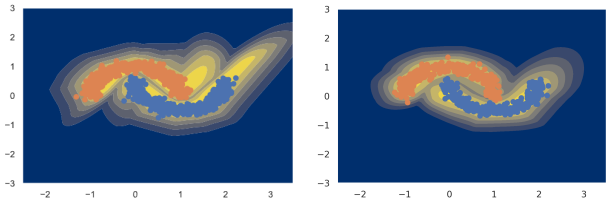
- The proposed technique is simple and seems easy to implement and apply.

2.3.3. Weaknesses

- The authors do not conduct experiments to evaluate uncertainty calibration, which is important for estimating the validness of given uncertainty.

- The two-sided gradient penalty can be undesirable for a residual network, since imposing $\nabla f(x) = 1$ onto a residual connection $f(x) = x + g(x)$ can force g(x) toward 0, leading to an identity mapping.

- The paper claims that two-sided gradient penalty can behave better than a single-sided penalty[1] by showing empirical results without further explanation, which may confuse readers. [2]



(a) DUQ reproduced by us (b) DUQ shown in paper

Figure 2. Two Moons Experiment

3. Experimental Reproduction

In this section, we conduct experiments mentioned in the investigated paper[3]. Specifically, we first validate the methods on the toy dataset, two-moons. Then, two pairs of OoD datasets are explored: FashionMNIST vs MNIST, and CIFAR-10 vs SVHN. Our experimental results verify the reproducibility of the paper. In section 3.4, we summarize the limitations are potential directions for future work.

3.1. Two Moons

The paper uses the scikit-learn(Pedregosa et al., 2011) implementation of this dataset. We follow it to set the noise level to 0.1 and generate 1000 points for the training set. Other settings are kept the same with the investigated paper. In Figure 2, we compare the visualization reported in the paper and reproduced by us. The brighter area represents a higher kernel value, and lower predictive uncertainty. Overall, both of the visualizations demonstrate high uncertainty for areas far from the distribution and areas near the decision boundary. However, our visualization is less satisfactory in terms of sensitivity capabilities compared to the reported results. See there are some OoD data with low uncertainty on the right of the data distribution. This is probably caused by the unstable properties of gradient penalty, and worth further investigation and improvements.

3.2. FashionMNIST vs MNIST/NotMNIST

One task of testing the quality of uncertainty estimation algorithms is out-of-distribution(OoD) detection, where some

[1]The one-sided penalty:$\lambda \cdot \max\left(0, \left\|\nabla_x \sum_c K_c\right\|_F^2 - 1\right)$

[2]Here is our intuitive explanation, which we hope will help to understand: The one-sided penalty has a broader limit between 0 and 1, which may not encourage sensitivity..

[3]We build the reproduction experiments on the officially released code, available at https://github.com/y0ast/deterministic-uncertainty-quantification.

	reproduction	reported
Acc	0.926	0.924 ± 0.2
AUROC MNIST	0.940	0.955 ± .007
AUROC NotMNIST	0.950	0.946 ± .018

Table 1. Comparision of our reproduction results and the reported results in FashionMNIST vs MNIST/NotMNIST.

	reproduction	reported
Acc($\lambda = 0$)	0.941	0.942 ± 0.2
AUROC($\lambda = 0$)	0.876	0.861 ± 0.032
Acc($\lambda = 0.5$)	0.934	0.932 ± 0.4
AUROC($\lambda = 0.5$)	0.916	0.927 ± 0.013

Table 2. Comparision of our reproduction results and the reported results in CIFAR-10 vs SVHN.

OoD data are used at test time to assess the ability of trained models to be able to separate them from in-distribution data. In this section, the authors use FashionMNIST(Xiao et al., 2017) as training set, and evaluate at two OoD dataset: MNIST(LeCun, 1998) and NotMNIST(Bulatov, 2011). It is stated that Fashion MNIST vs MNIST is a more challenging task than Fashion MNIST vs NotMNIST.

For evaluation, besides the normal metrics of accuracy on the in-distribution data [4], a specific metrics for OoD detection is used for evaluating model ability to reject OoD data, It's expected that model gives low uncertainty for Fashion-MNIST test set, high uncertainty to MNIST and NotMNIST dataset.[5] Therefore, we reject a portion of test data with relatively high uncertainty. And it is evaluated by AUROC value, with higher being better. Ideally, the metrics equal to 1 if the model can perfectly separate OoD data from original in-distribution test data.

For experiment settings, we set the length scale to 0.1, the gradient penalty λ to 0.05, which are reported to have the best performance in the original paper. Other settings are kept the same. In Table 1, we compare our reproduction experimental results with the reported ones[6]. As can be seen, our values are within the range of the original paper, which verifies the proposed results.

3.3. CIFAR-10 vs SVHN

In this section we conduct experiments on the CIFAR-10 dataset(Krizhevsky et al., 2009), with SVHN(Netzer et al., 2011) as OoD dataset.

Compared to the last dataset pairs, CIFAR-10 is more difficult for out of distribution detection for several reasons. Firstly, there is a significant amount of data noise: some of the dog and cat examples are not distinguishable using only 32 by 32 pixels. Secondly, the training set is small compared to its complexity, making it easy to overfit without data augmentation. So the paper uses random horizontal flips and random crops as data augmentation to prevent the model from overfitting.

We follow the original experimental settings, with the length scale being 0.1 and the gradient penalty λ being 0 and 0.5. Results are shown in Table 2, with metrics described above. As can be seen, our reproduction results are comparable to the reported results.

Compared to $\lambda = 0$, the authors find a slight decrease in the prediction accuracy at $\lambda = 0.5$ and a significant increase in AUROC. From this, the authors infer that without a gradient penalty, the model can slightly better adapt to the training data, resulting in higher accuracy, but due to the lack of sensitivity, it may project different data features to similar locations in the feature space and reduce the model's ability to detect uncertainty.

3.4. Future work

Our reproduction experiments verify the reproducibility of the paper. From the experiments, we also identify some limitations and potential directions for future work.

- From the visualization in the Two Moons dataset, we find the gradient penalty method can be unstable. Also, in the methodology review, we explain its limitation on the residual block. For these, we encourage future works on finding more stable and effective ways to improve the sensitivity ability of the model.

- For some difficult datasets, such as CIFAR-10, the OoD detection can be affected by ambiguity within the in-distribution data, which are known to have high *aleatoric uncertainty*. Currently, the DUQ method can not distinguish in-distribution data with high aleatoric uncertainty. For this, we do some modifications to the original DUQ method and conduct experiments on these three datasets, which we will explain in detail in the next section.

4. Further Exploration

4.1. Motivation

We begin with a discussion of two types of uncertainty, namely *epistemic uncertainty* and *aleatoric uncertainty*. Epistemic uncertainty measures how well the model is matched to the data. Aleatoric uncertainty, on the other hand, arises from the natural complexity of the data.

[4] Here, the in-distribution data means the data from FashionMNIST, but not be seen during training

[5] since the model has never seen these datasets before and they are very different from FashionMNIST.

[6] We store all the product model files in our submitted code.

The out-of-distribution data is supposed to have high epistemic uncertainty. However, the uncertainty given by the DUQ method contains both epistemic uncertainty and aleatoric uncertainty, making it hard to distinguish OoD data with high epistemic uncertainty from in-distribution data with high aleatoric uncertainty.

Motivated by this, we aim to separate data with high aleatoric uncertainty and data with high epistemic uncertainty, to further improve the model's performance in OoD detection.

4.2. Generalized DUQ

We name our tiny improvement version as Generalized DUQ (G-DUQ), where we keep the training procedure and only extend the uncertainty estimation criterion in inference time. Intuitively, data with high aleatoric tends to sit between centroids and OoD data is always far away from all centroids. Imagine a specific case, where point A and point B both have the same closest distance, while point A is between two centroids, and point B is only close to one centroid and far away from others. In this situation, point A is more potential to have high aleatoric uncertainty, and point B is more likely to have high epistemic uncertainty. However, the original DUQ method using the closest distance can not separate the two cases.

Therefore, we extend the original uncertainty estimation of only utilizing its nearest centroid to utilize the nearest k centroids ($k \leq$ class number). Specifically, the uncertainty is obtained from the mean of the uncertainty of the top k closest centroids:

$$U(\mathbf{x}, k) = \frac{1}{k} \sum_{c \in N(k,x)} K_c \left(f_\theta(\mathbf{x}), \mathbf{e}_c \right) \tag{7}$$

where $N(k, x)$ represents the class set with corresponding centroids being k-nearest.

$$N(k, x) = \arg\max_{|N(k,s)|=k} \sum_{c \in N(k,x)} K_c \left(f_\theta(\mathbf{x}), \mathbf{e}_c \right) \tag{8}$$

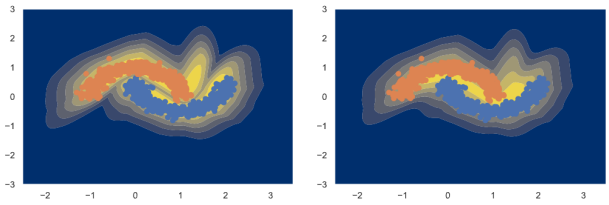
When k=1, the G-DUQ degenerate to original DUQ.

4.3. Experiments

For the following experiments, we first use the toy dataset Two Moons to visualize the changes in uncertainty, followed by testing the performance of G-DUQ on the two OoD dataset pairs. For each real dataset, we do a series of experiments for different k for ablation study.

4.3.1. Two Moons

In Figure 3. we compare the Generalized DUQ (G-DUQ) with k = 2, to the original DUQ method. It can be seen that the G-DUQ reduces uncertainty near the boundary, where data in this region is known to have large aleatoric uncertainty. This visualization evident our assumption that G-DUQ can separate data with high epistemic uncertainty from data with high aleatoric uncertainty.



(a) Original DUQ (b) Generalized DUQ

Figure 3. Comparision of DUQ and G-DUQ on Two Moons.

	k=1	k=2	k=3	k=4
AUROC MNIST	0.94	0.97	0.97	0.97
AUROC NotMNIST	0.95	0.97	0.98	0.98

Table 3. Comparision of DUQ and G-DUQ on FashionMNIST vs MNIST/NotMNIST.

4.3.2. FashionMNIST vs MNIST/NotMNIST

We evaluate the performance of G-DUQ on the OoD dataset pairs and set k to different values. Results are shown in Table 3, with $k = 1$ being the results of the original DUQ method.[7] Here we only compare the changes in AUROC, since accuracy will remain the same as Table 1. [8]

From the results, we can see that this modification does benefit the performance and increases AUROC for both settings. We also notice that with the increase of k, AUROC increases and saturates. This may due to the centroids in high-dimensional feature space are close to each other.

4.3.3. CIFAR-10 vs SVHN

Similarly, we set k to different values and observe the change of AUROC for separating OoD data in Table 4.

Results are somewhat different from before. We find that with the increase of k, AUROC tends to decrease. We assume that this phenomenon might be caused by the complexity of the dataset where data from different classes are significantly different from each other. In this case, centroids might be far from each other resulting in a sparse feature space, and increasing k does not contribute to the performance.

[7] We use the reproduction value here, and there is no much difference between reported values as discussed before.

[8] We only change the criterion of evaluating uncertainty, and that will not affect the model's prediction.

	k=1	k=2	k=3	k=4
AUROC	0.916	0.864	0.753	0.644

Table 4. Comparision of DUQ and G-DUQ on CIFAR-10 vs SVHN.

5. Conclusion

In this report, we investigate a paper in the uncertainty estimation domain. We introduce the main ideas of the method, analysis its contribution, strengths, and weaknesses. We also reconduct the experiments and find some limitations empirically. For one of those limitations, we design a simple strategy to solve it. Experiments on Two Moons and FashionMNIST vs MNIST/NotMNIST validate its improvement. On CIFAR-10 vs SVHN, the experiment results are less satisfactory, which we give a potential explanation and leave to future work.

References

Bulatov, Y. Notmnist dataset. *Google (Books/OCR), Tech. Rep.[Online]. Available: http://yaroslavvb. blogspot. it/2011/09/notmnist-dataset. html*, 2, 2011.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. *arXiv preprint arXiv:1602.04621*, 2016.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning*, pp. 9690–9700. PMLR, 2020.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.