M.S. THESIS

Robustness of Neural Networks under Distributional Shifts

분포 변화 상황에서 뉴럴 네트워크의 강건성

BY

JUHONG SONG

AUGUST 2021

Interdisciplinary Program
in Computational Science and Technology
Seoul National University

Robustness of Neural Networks under Distributional Shifts

분포 변화 상황에서 뉴럴 네트워크의 강건성

지도교수 강 명 주

이 논문을 이학석사 학위논문으로 제출함

2021년 4월

서울대학교 대학원

협동과정 계산과학전공

송 주 홍

송주홍의 이학석사 학위 논문을 인준함

2021년 5월

위 원 장: _____

부위원장: _____

위　　원: _____

Robustness of Neural Networks under Distributional Shifts

A thesis
submitted in partial fulfillment
of the requirements for the degree of
Master of Science
to the faculty of the Graduate School of
Seoul National University

by

JUHONG SONG

Thesis Director : Professor Myungjoo Kang

Interdisciplinary Program
in Computational Science and Technology
Seoul National University

August 2021

Abstract

In this thesis, uncertainty estimation is performed under distributional shifts. The goal of uncertainty estimation is to create reliable deep learning models which can yield a confidence value with its prediction. Although several studies have been conducted to quantify uncertainty in the deep learning models, recent studies have demonstrated that the quality of uncertainty estimated using some traditional methods degrades in dataset shift situations. In this paper, we propose Contrastive Normalizing Flow, a robust uncertainty estimation model under distributional shifts. The proposed model estimates uncertainty in a latent space; An encoder trained with contrastive learning maps images into the latent space. Then, a generative classifier models a predictive distribution with normalizing flows. In addition to this, distributionally robust optimization is applied to the proposed model to improve a performance of out-of-distribution detection. Two types of shifts are considered in experiments: covariate shift and out-of-distribution. For these types of shifts, the experiments empirically demonstrate that the proposed model improves the robustness of the classifier under distributional shifts.

keywords: Out of distribution, robustness, uncertainty

student number: 2019-29928

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The ultimate goal of artificial intelligence (AI) is to create human-level intelligence, and one of the key features of human-level intelligence is generalization. Modern research directions in AI have been made mainly by using deep learning, because it has achieved great success in several areas such as image recognition and machine translation. However, one of the main problems of neural networks is that it lacks generalization ability: the networks tend to overfit to train data and find it difficult to predict unseen data. To overcome this limitation, many researchers have attempted to give the networks a generalization property for various tasks.

1.1 Motivation

Classification is one of the tasks that the researchers have struggled to generalize. What is the meaning of generalization in classification? There are two implications. First, the neural networks should be robust under covariate shift. Suppose I train the network with training distribution $p(x, y)$. Then, covariate shift indicates that $p(x)$ becomes different with the same conditional distribution $p(y|x)$. As test distribution is usually different from training distribution up to $p(x)$, the networks typically face covariate shift.

The second implication is that the networks should predict what they do not know. That means if the networks receive completely different data from the training distribution, then they should yield low confidence with their predictions. This situation is called out-of-distribution (OOD). In classification, confidence is usually estimated by maximum softmax probability over the class. However, Guo et al. [6] revealed that modern neural networks are likely to provide overconfidence of their predictions. In other words, the networks incorrectly classify data with high confidence. This leads to a requirement to adjust the network's confidence more reliably, and this process is called *calibration*.

One of the research directions that attempts to calibrate the prediction's confidence is uncertainty estimation. The word 'uncertainty' here means uncertainty about the prediction. As mentioned above, modern neural networks tend to yield unreliable uncertainty estimates. Therefore, many researches on uncertainty have aimed to build a model that yields more well-calibrated uncertainty.

Traditional methods to quantify uncertainty are divided into the two categories: Bayesian and non-bayesian neural networks. Bayesian neural networks (BNNs) assumes that the parameters of the networks follow probability distributions, and estimates a posterior distribution over the parameters. As the marginalization of likelihood is required to inference posterior distribution over the parameters, the exact posterior cannot be evaluated. This leads to a study of BNN in the direction of how to approximate the posterior accurately while reducing the computational cost. Stochastic variational inference (SVI) approximates the posterior using Monte Carlo estimates with an assumption that the posterior follows a diagonal Gaussian distribution [9]. SVI allowed the Bayesian method to be applied to large-scale networks, which became the basis of subsequent BNN methods, such as Probabilistic Back Propagation [8], BBB [1], and Monte Carlo dropout [5].

Although these methods appear to yield reliable uncertainty estimates, a recent research demonstrated that quality of uncertainty estimated by traditional methods such

as Dropout [5], Ensemble [13], and BBB [1] tend to degrade in distributional shift situations [19]. In other words, the prediction confidences are likely to be overconfident under a covariate or OOD shift. Especially, a reliable uncertainty estimation is more important in real-world situation where the model typically receives the data different from those of the training distribution. Therefore, recent researches to quantify uncertainty have focused on ensuring a robustness of a neural network under distributional shifts.

One of the researches that attempts to solve this problem employs distributional shift data during the training phase for the purpose of reducing a prediction confidence over the shifted data. Malinin et al. [16] modeled a predictive distribution using the Dirichlet distribution. They trained the model in a way that the training distribution got closer to an unsymmetric Dirichlet distribution, and OOD approached closer to a flat Dirichlet distribution. Then the resulting model was able to yield a high and low confidence for the training and OOD data, respectively. However, this method requires OOD data during the training phase to derive the predictions of OOD input with low confidence, which is not always possible. Moreover, as all data except the training data corresponds to OOD data, it is desirable that the resulting model yields a low confidence for all data remote from the training distribution. However, it may be possible that a robustness of the resulting model over OOD cannot generalize beyond the provided OOD data. In other words, the prediction confidence can be overconfident to OOD data, not provided in the training phase.

Posterior Network (PostNet) [3] solved the above problem by using an encoder and normalizing flows. They assumed that a predictive distribution follows the Dirichlet distribution, and updated the concentration parameters of the Dirichlet distribution based on a density value $p(z \mid y = c)$ estimated by normalizing flows. Their motivation is that the encoder would embed examples belonging to the same class into close latent positions in the latent space, and the flow would assign the examples to higher density values if the corresponding latent positions are close to the clusters generated

by the encoder. It follows that OOD examples are naturally assigned to a low density value due to a far distance between the latent positions of OOD examples and clusters. However, a recent research [17] demonstrated that normalizing flows failed to detect OOD data. They empirically showed that a flow-based model tended to assign a higher likelihood to OOD data than training one. To solve this problem, [12] trained a flow-based model with OOD data such that the model assigned a low likelihood to provided OOD data. Although their method succeeded in yielding the desired results, their scope was restricted within the provided OOD data, the same as [16].

1.2 Contribution

The objective of this thesis is to generalize classification such that a trained network is robust under a covariate and OOD shift. The proposed method is extended from PostNet in a sense that a basic structure of the proposed model is the same as PostNet, but the proposed model overcomes some limitations of PostNet.

First, contrastive learning is employed to train an encoder as a means of increasing a clustering performance of the encoder. The better clustering, the more likely the similar inputs are to be lumped together in the latent space and the further OOD inputs are to be moved away from the clusters. It follows that the better clustering leads to a more reliable uncertainty estimation.

Second, the Dirichlet distribution is adopted as a base distribution of conditional normalizing flow to increase an expressivity of a flow-based model. As in PostNet, the role of the proposed normalizing flow is to estimate a conditional density $p(z \mid y = c)$. We increase expressivity of the conditional normalizing flow by replacing the base distribution from the Gaussian mixture to the Dirichlet distribution, which leads to estimating more accurate conditional density. An ablation study will demonstrate that the Dirichlet based normalizing flow improves calibration upon the Gaussain mixture based normalizing flow.

Finally, distributionally robust optimization (DRO) is used to train the proposed model to improve OOD detection performance. With DRO, the proposed model can detect OOD data even if the model have not seen those data in the training phase. An ablation study will demonstrate that the proposed DRO method can improve OOD detection performance without provided OOD data during training.

Chapter 2

Related Works

2.1 Supervised Contrastive Learning

Contrastive learning is a learning framework in which similar inputs are mapped into close areas in the latent space [4]. In a supervised setting, contrastive learning assumes that the inputs from the same categories are similar; hence the trained model places the same class together in the latent space [11]. Let $E : \mathcal{X} \rightarrow \mathcal{Z}$ be an encoder that maps the inputs to the latent space. Then for each iteration, the minibatch set $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ is independently sampled from the training distribution $P(\mathbf{x}, y)$. Next, $\mathcal{D}$ is augmented with two random transformations t_1 and t_2 obtained from the augmentation distribution $\mathcal{T}$, thereby resulting in an augmented minibatch set $\tilde{\mathcal{D}} = \{\tilde{\mathbf{x}}_i, \tilde{y}_i\}_{i=1}^{2n}$ where $\tilde{\mathbf{x}}_{2i-1} = t_1(\mathbf{x}_i), \tilde{\mathbf{x}}_{2i} = t_2(\mathbf{x}_i)$, and $\tilde{y}_{2i-1} = \tilde{y}_{2i} = y_i$ for $i = 1, 2, ..., n$. Subsequently, the following objective is minimized over $\tilde{\mathcal{D}}$ as follows:

$$\ell_{CT}(\mathbf{x}, y) = -\sum_{\substack{j=1 \\ i \neq j}}^{2N} \mathbb{1}_{[\boldsymbol{y}_i = \boldsymbol{y}_j]} \cdot \log \frac{\exp\left(\mathbf{z}_i \cdot \mathbf{z}_j / \tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[i \neq k]} \cdot \exp\left(\mathbf{z}_i \cdot \mathbf{z}_k / \tau\right)}, \tag{2.1}$$

where $\mathbf{z} = E(\mathbf{x})$. The indicator function $\mathbb{1}_{[\boldsymbol{y}_i = \boldsymbol{y}_j]}$ ensures that the numerators of the log arguments represent the similarity between the latent variables belonging to the same class. This induces the networks to learn parameters such that the inputs are clustered based on their classes.

2.2 Normalizing Flows

Normalizing flows can express a complex probability distribution using multiple transformations from a simple distribution. One of the advantages of a flow-based model is that it can evaluate an exact likelihood, thereby enabling an exact density evaluation. Let $\mathbf{x} \in \mathbb{R}^d$ be a continuous random variable with density $p_{\mathcal{X}} : \mathbb{R}^d \to [0, \infty)$. Normalizing flows is defined with invertible and differentiable transformations $f_i : \mathbb{R}^d \to \mathbb{R}^d$, and its marginal likelihood $p_{\mathcal{X}}(\mathbf{x})$ is evaluated as follows:

$$\log p_{\mathcal{X}}(\mathbf{x}) = \log p_{\mathcal{Z}}(\mathbf{z}) + \Sigma_{i=1}^{n} \log \left| \det \frac{\partial f_i(\mathbf{z}_i)}{\partial \mathbf{z}_i} \right|, \tag{2.2}$$

where $\mathbf{z}_i = f_i(\mathbf{z}_{i-1}), \mathbf{z}_0 = \mathbf{x}, \mathbf{z}_n = \mathbf{z}$, and $p_{\mathcal{Z}}(\mathbf{z})$ is a base distribution of normalizing flows. As the above equations, the evaluation of the likelihood requires the Jacobian determinant of the transformations, which restricts the allowable transformations within invertible and differentiable functions. The bijective transformations limit an expressivity of the flows; a flow-based model cannot modify the dimension of the variables during the transformations and other techniques are required to model discrete data.

2.3 Variational Autoencoder

A variational autoencoder (VAE) is also a generative model to express a probability distribution with a neural network. Unlike a flow-based model, VAE can learn a posterior distribution over the latent variables with a dimension reduction. As the true posterior is intractable, it approximates the posterior with a variational distribution. This leads to an objective of VAE becoming evidence lower bound as follows:

$$\begin{aligned} \log p_{\mathcal{X}}(\mathbf{x}) &= \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{z})] + \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x} \mid \boldsymbol{z})}{q(\boldsymbol{z} \mid \boldsymbol{x})}\right] + \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{q(\boldsymbol{z} \mid \boldsymbol{x})}{p(\boldsymbol{z} \mid \boldsymbol{x})}\right] \\ &\geq \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x} \mid \boldsymbol{z})] - \mathbb{D}_{\mathrm{KL}}[q(\boldsymbol{z} \mid \boldsymbol{x}) \| p(\boldsymbol{z})], \end{aligned} \tag{2.3}$$

where $\mathbb{D}_{KL}$ represents a KL-divergence. While VAE can reduce the dimension of the input space, its objective is a lower bound of the marginal likelihood, which does not guarantee a global optimal solution.

2.4 SurVAE Flows

Surjective VAE Flows (SurVAE Flows) is a framework to link a gap between normalizing flows and VAE using surjective transformations [18]. As a surjective transformation is not invertible, the framework replaces an inverse function with a stochastic transformation. The two types of transformations are considered in the framework: generative surjection whose forward function $(z \to x)$ is a deterministic surjective function $(x = f(z))$ and backward function $(x \to z)$ is a stochastic transformation $(z \sim q(z \mid x))$, and inference surjection whose forward function is a stochastic transformation $(x \sim p(x \mid z))$, and backward function is a deterministic surjective function $\left(z = f^{-1}(x)\right)$. For inference surjection, marginal likelihood can be exactly calculated as follows:

$$\log p_{\mathcal{X}}(\mathbf{x}) = \log p_{\mathcal{Z}}(\mathbf{z}) + \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x} \mid \mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})}\right]. \tag{2.4}$$

Using these transformations, the author proved that Equation (2.2) can be obtained from the first line of Equation (2.3). It implies that SurVAE Flows framework can bridge the difference between VAE and normalizing flows. Under this framework, a surjective transformation can be used as a component of normalizing flows, thereby allowing to use functions unavailable in normalizing flows such as dimension reduction and quantization. This leads to increasing an expressivity of a flow-based model.

Chapter 3

Methods

This chapter describes a proposed method to quantify uncertainty. Unlike other uncertainty estimation models, the proposed method estimates uncertainty in the latent space.

3.1 Uncertainty in the latent space

To do this, an input data is first mapped to a corresponding latent position through an encoder trained with supervised contrastive learning Equation (2.1). Then the resulting encoder maps the inputs belonging to the same class into closed areas in the latent space, which generates clusters for each class. As the encoder produces the clusters based on the training distribution, it is likely to embed the shifted inputs into the latent positions away from the clusters. Therefore, the proposed model estimates uncertainty based on the distance from the clusters. In other words, the model assigns a low confidence with its prediction if the latent position of the input is far from the clusters, and a high confidence if the position is close to the clusters. The distance from the clusters can be estimated through a conditional density $p_{\mathcal{Z}}(\mathbf{z}|y = k)$, which has a high value at the latent position where kth class examples are gathered. Therefore, normalizing flow is used to evaluate the conditional density $p_{\mathcal{Z}}(\mathbf{z}|y = k)$ explicitly.

3.2 Dirichlet Normalizing Flows

Normalizing flows can be employed for a classification by using a conditional density in the Equation (2.2) as follows:

$$\begin{aligned}\mathcal{L}_\phi(\mathbf{z}, y) &= \log p_{\mathcal{Z}}(\mathbf{z}|y=k;\phi) \\ &= \log p_{\mathcal{W}}(\mathbf{w}|y=k) + \sum_{i=1}^{n} \log \left| \det \frac{\partial f_i(\mathbf{z}_i;\phi_i)}{\partial \mathbf{z}_i} \right|.\end{aligned} \tag{3.1}$$

Previous researches on conditional normalizing flows use the Gaussian mixture as a base distribution of the flows [10]. However, the Gaussian mixture can only represent a symmetric mode of the data, thereby restricting the expressivity of a generative model [2]. Therefore, we use the Dirichlet distribution as a base distribution of the flows such that the proposed flows can represent an asymmetric mode as well as a symmetric mode, which increases a flexibility of the flow-based model and evaluates more accurate the conditional density in the latent space.

However, there is a constraint that the support of the Dirichlet distribution **resides in** a simplex, $\Delta^{D-1} = \left\{ \mathbf{w} \mid \sum_{i=1}^{D} w_i = 1 \text{ and } 0 \leq w_1, \ldots, w_D \leq 1 \right\}$. Therefore, the proposed flow-based model is added with a positive increasing function and normalization at the last layer of the flows. The problem is that a normalization operation is a non-invetible function such that it cannot be used as a component of normalizing flows. This motivates us to design a normalization layer in the SurVAE Flows framework, which enables to utilize a normalization operation in the flow-based model.

Suppose that positive random variables $\mathbf{v}$ are given. Then, the following operation can be used for normalizing $\mathbf{v}$ as a component of normalizing flows:

$$\begin{aligned}q(\mathbf{w} \mid \mathbf{v}) &= \delta\left(\mathbf{w} - \frac{\mathbf{v}}{sum(\mathbf{v})}\right) \\ p(\mathbf{v} \mid \mathbf{w}) &= \int p(\mathbf{v} \mid \mathbf{w}, s) p(s \mid \mathbf{w}) ds = \int \delta(\mathbf{v} - s\mathbf{w})\, p(s \mid \mathbf{w}) ds,\end{aligned} \tag{3.2}$$

where π is one layer network with an activation function with the parameters ψ and $sum(\mathbf{v}) = \Sigma_{i=1}^{D} v_i$. Equation (3.2) indicates that during the density estimation, $\mathbf{v}$ is

deterministically transformed into $\mathbf{w}$ with $\mathbf{w} = \frac{\mathbf{v}}{sum(\mathbf{v})}$, and for the generative process, $\mathbf{w}$ is stochastically transformed into $\mathbf{v}$ with $\mathbf{v} = s\mathbf{w}$ where s is sampled from $\mathrm{Gamma}(\pi_\psi(\mathbf{w}), 1)$. This normalization layer can be incorporated into normalizing flows by replacing one of the log determinant terms in Equation (3.1) as follows:

$$\begin{aligned} \log p_{\mathcal{Z}}(\mathbf{z}|y=k;\phi) = \log p_{\mathcal{W}}(\mathbf{w}|y=k) + \sum_{i=1}^{n-1} \log \left| \det \frac{\partial f_i\left(\mathbf{z}_i;\phi_i\right)}{\partial \mathbf{z}_i} \right| \\ + \log p_{\mathcal{S}}(s \mid \mathbf{w}) - D \log\left(s\right), \end{aligned} \tag{3.3}$$

where $D = |\mathcal{Y}|, s = sum(\mathbf{v})$.

Here, we derive the specific derivation of the Equation (3.3). Suppose random variables $\mathbf{z}$ are transformed into positive random variables $\mathbf{v}$ by using $n-1$ functions of normalizing flows. Then, the conditional density $p_{\mathcal{Z}}(\mathbf{z}|y=k)$ is calculated as follows:

$$\log p_{\mathcal{Z}}(\mathbf{z}|y=k;\phi) = \log p_{\mathcal{V}}(\mathbf{v}|y=k) + \sum_{i=1}^{n-1} \log \left| \det \frac{\partial f_i\left(\mathbf{z}_i;\phi_i\right)}{\partial \mathbf{z}_i} \right|. \tag{3.4}$$

Subsequently, $\mathbf{v}$ is transformed into the Dirichlet random variables using the proposed normalization layer, and the conditional density $\log p_{\mathcal{V}}(\mathbf{v}|y=k)$ is calculated by Equation 2.4:

$$\log p_{\mathcal{V}}(\mathbf{v} \mid y=k) = \log p_{\mathcal{W}}(\mathbf{w} \mid y=k) + \mathbb{E}_{q(\mathbf{w}|\mathbf{v})}\left[\log \frac{p(\mathbf{v} \mid \mathbf{w})}{q(\mathbf{w} \mid \mathbf{v})}\right]. \tag{3.5}$$

Surjection	**Forward**	**Inverse**	**Likelihood Contribution**
Norm	$\mathbf{v} = s\mathbf{w}$ where $s \sim \mathrm{Gamma}(\pi_\psi(\mathbf{w}), 1)$	$\mathbf{w} = \frac{\mathbf{v}}{sum(\mathbf{v})}$	$\log p(s\|\mathbf{w}) - D\log(s)$

Table 3.1: Proposed Normlization layer in the SurVAE Framework

Then, the expectation term in Equation (3.5) is calculated as follows:

$$\mathbb{E}_{q(\mathbf{w}|s,\mathbf{v})q(s|\mathbf{v})}\left[\log \frac{p(\mathbf{v} \mid s, \mathbf{w})p(s \mid \mathbf{w})}{q(\mathbf{w} \mid s, \mathbf{v})q(s \mid \mathbf{v})}\right] = \int q(\mathbf{w}, s \mid \mathbf{v}) \log \frac{p(\mathbf{v} \mid s, \mathbf{w})p(s \mid \mathbf{w})}{q(\mathbf{w} \mid s, \mathbf{v})q(s \mid \mathbf{v})} d\mathbf{w}ds$$

$$= \int q(\mathbf{w}, s \mid \mathbf{v}) \log \frac{\delta(\mathbf{v} - s\mathbf{w})p(s \mid \mathbf{w})}{\delta(\mathbf{w} - \frac{\mathbf{v}}{s})\delta_{s,sum(v)}} d\mathbf{w}ds$$

$$= \int q(\mathbf{w}, s \mid \mathbf{v}) \log \frac{p(s \mid \mathbf{w})}{s^D \cdot \delta_{s,sum(v)}} d\mathbf{w}ds \quad \left(\because \delta(\mathbf{v} - s\mathbf{w}) = \delta\left(\mathbf{w} - \frac{\mathbf{v}}{s}\right) \cdot \frac{1}{|s|^D}\right)$$

$$= \int q(\mathbf{w} \mid s, \mathbf{v})q(s \mid \mathbf{v}) \log \frac{p(s \mid \mathbf{w})}{s^D \cdot \delta_{s,sum(v)}} d\mathbf{w}ds$$

$$= \int \delta\left(\mathbf{w} - \frac{\mathbf{v}}{s}\right) \delta_{s,sum(\mathbf{v})} \log \frac{p(s \mid \mathbf{w})}{s^D \cdot \delta_{s,sum(v)}} d\mathbf{w}ds$$

$$= \int \delta_{s,sum(\mathbf{v})} \log \frac{p(s \mid \mathbf{w})}{s^D \cdot \delta_{s,sum(v)}} ds \quad \text{where} \quad \mathbf{w} = \frac{\mathbf{v}}{s}$$

$$= \int \delta_{s,sum(\mathbf{v})} \log p(s \mid \mathbf{w}) - \delta_{s,sum(\mathbf{v})} \log\left(s^D \cdot \delta_{s,sum(v)}\right) ds$$

$$= \log p(s \mid \mathbf{w}) - D \log(s) \quad \text{where} \quad s = sum(\mathbf{v})$$

By substituting Equation (3.5) for $\log p_{\mathcal{V}}(\mathbf{v}|y = k)$ in Equation (3.4), the Equation (3.3) is obtained. Table 3.1 summarizes the proposed normalization layer.

In summary, random variables $\mathbf{z}$ are transformed into positive random variables $\mathbf{v}$ by using normalizing flows, and the resulting $\mathbf{v}$ is transformed into the Dirichlet random variables $\mathbf{w}$ by using the proposed normalization layer. For concentration parameters, we assign higher values to the kth dimension of α_k. The parameter settings of the conditional normalizing flows are described in the experimental section.

Until now, the Dirichlet normalizing flow is trained by maximizing $\log\, p_{\mathcal{Z}}(\mathbf{z}|y = y_{true})$ over the training distribution. As the encoder is trained by mapping the training distributions into near the clusters, the flow is likely to yield high density values for data near the clusters. However, under distributional shifts, the encoder is likely to

place the shifted data away from the clusters, because the encoder has not seen those data. The flow is trained by maximum likelihood over the examples near the clusters such that it is expected to assign a lower likelihood to the examples far from the clusters. However, recent studies [12, 17] demonstrated that a flow-based model was likely to assign higher likelihood to OOD data. Kirichenko et al. [12] mitigated this problem by using OOD dataset during the training phase in a way that maximized likelihood over the training data and minimized likelihood over OOD data. Although the resulting flow can distinguish between the training data and OOD data, an additional data is required to give the flow an ability to distinguish, and the resulting flow could only detect the provided OOD data. We overcome this limitations without an additional data in the training phase. With the assumption that we possess only training data, our goal is to enhance OOD detection performance without using OOD data. This motivates us to consider distributionally robust optimization for OOD detection.

3.3 Distributionally Robust Optimization for OOD detection

The objective for this part is to reduce the prediction's confidence of OOD data. As the encoder is likely to embed OOD data into a latent position away from the clusters, all we have to do is to encourage the model to yield low confidence over the examples far from the clusters. This requires the examples away from the clusters during the training phase, but we only have the training data. Therefore, we virtually generate such examples by using distributionally robust optimization (DRO) and use those examples during the training phase.

Suppose that a Wasserstein ball of radius $\rho > 0$ centered at $P(\mathbf{x})$ is given as follows:

$$\mathcal{B}_r(P) = \{Q : W_c(P, Q) \leq \rho\}, \tag{3.6}$$

where $W_c(\cdot, \cdot)$ is a Wasserstein distance between two probability distributions with a transportation cost c. Then, the worst-case distribution within the ball can be obtained

by solving the following DRO problem:

$$\inf_{Q \in \mathcal{B}_r(P)} \mathbb{E}_Q \left[\log p \left(E_\theta(\mathbf{x}) \mid y = k \right) \right], \tag{3.7}$$

where $P(\mathbf{x})$ is the training distribution. The solution of the above optimization problem is a shifted distribution $Q(\mathbf{x})$ whose sample is embedded into a latent position that has the lowest values of $p(\mathbf{z}|y = k)$ within $\mathcal{B}_r(P)$. It indicates that the samples from $Q(\mathbf{x})$ are away from the clusters. If the model is trained in a way that a prediction entropy over the samples from $Q(\mathbf{x})$ is increased, the resulting flow yields low confidence predictions over the examples far from the clusters.

As the Equation (3.7) is intractable, the DRO problem is relaxed by using the Proposition 1 from [21], and the relaxed optimization probelm is as follows:

$$\inf_{Q} \mathbb{E}_Q \left[\log p \left(E_\theta \left(\mathbf{x}' \right) \mid y = k \right) + \gamma c \left(\mathbf{x}', \mathbf{x} \right) \right], \tag{3.8}$$

where $\mathbf{x} \sim P(\mathbf{x}), \mathbf{x}' \sim Q(\mathbf{x}), \gamma$ is a positive relaxation parameter and $c(\cdot, \cdot)$ is the ℓ_2 distance. Suppose that a mini-batch $\{\mathbf{x}_i\}_{i=1}^n$ is sampled from the training distribution $P(\mathbf{x})$. Then, the approximate solutions of the Equation (3.8) can be obtained by using the following iterations:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t - \eta \cdot \nabla_{\mathbf{x}} \left[\log p \left(E_\theta \left(\mathbf{x}_i^t \right) \mid y = y_i \right) + \gamma c \left(\mathbf{x}_i^t, \mathbf{x}_i^0 \right) \right], \tag{3.9}$$

where $\mathbf{x}_i^0 = \mathbf{x}_i$ and η is a step size. If T iterations are performed, the resulting $\left\{ \mathbf{x}_i^T \right\}_{i=1}^n$ are approximate samples from $Q(\mathbf{x})$ whose Wasserstein distance from $P(\mathbf{x})$ is within ρ. Subsequently, the prediction's entropy over those examples can be increased by maximizing the following objectives:

$$H_\phi(\mathbf{z}) = \mathbb{E}_{Q(\mathbf{x})} \left[- \log p_{\mathcal{Z}}(\mathbf{z} \mid y; \phi) \right], \tag{3.10}$$

where $\mathbf{z} = E_\theta(\mathbf{x})$. The learning algorithm is summarized in Algorithm 1.

Algorithm 1 Pseudocode for the learning procedure

$\ell_{CT}(\cdot)$ is supervised contrastive loss (1)

$\mathcal{L}_\phi(\cdot)$ is conditional log likelihood (4)

$H_\phi(\cdot)$ is entropy (12), $F_\phi(\cdot)$ is normalizing flow

Initialize θ, ϕ, ψ

for m = 1, 2, ..., **do**

    Sample $\{\mathbf{x}_i, y_i\}_{i=1}^n$ from $p(\mathbf{x}, y)$

    if m $\leq$ threshold **then**

        Minimize $\frac{1}{n}\sum_{i=1}^n \ell_{CT}\left(E_\theta(\mathbf{x}_i), y_i\right)$

    else

        Generate $\{\mathbf{x}'_i\}_{i=1}^n$ with (11)

        Maximize $\frac{1}{n}\sum_{i=1}^n \mathcal{L}_\phi\left(\mathbf{z}_i, y_i\right) + H_\phi(\mathbf{z}'_i)$ where $\mathbf{z}_i = E_\theta(\mathbf{x}_i), \mathbf{z}'_i = E_\theta(\mathbf{x}'_i)$

        Maximize $\frac{1}{n}\sum_{i=1}^n \log p_\psi(s_i|\mathbf{w}_i)$ where $s_i = \pi_\psi(\mathbf{w}_i), \mathbf{w}_i = F_\phi(\mathbf{z}_i)$

    end if

end for

Chapter 4

Experiments

In this chapter, robustness of the proposed model is evaluated under distributional shifts. The two types of shift are considered: a covariate shift (CIFAR-10-Corruption and MNIST Rotation) and OOD shift. Moreover, ablation studies are conducted to analyze the effect of the Dirichlet base distribution for calibration and the DRO for OOD detection performance. The proposed model is compared with the baselines: Dropout [5], Ensemble [13], BBB [1], and PostNet [3]. The rationale for the selection is that Dropout, Ensemble, and BBB are traditional methods to quantify uncertainty and the proposed model is inspired by PostNet, which is compared in the related works.

4.1 Evaluation Metric

1. **Expected calibration error** (ECE) measures the extent of matching between the confidence and accuracy. A low ECE indicates that the classifier is well calibrated, and yields reliable confidence with the predictions. Under covariate shift, we estimate the quality of the predictive uncertainty based on the ECE.

2. **FPR at 95% TPR** represents false positive rate (FPR) that a negative example (out-of-distribution) is incorrectly clas- sified as positive (in-distribution) when true positive rate (TPR) is about 95%. A low FPR implies that a model has an

ability to discriminate between a positive and negative example.

3. **AUROC** is the area under the ROC curve which shows a tradeoff between TPR and FPR for each threshold. If a model yields a large AUROC, then it can distinguish a positive example from a negative one robust to threshold values.

4. **AUPR** is the area under the Precision Recall curve which shows a tradeoff between precision and recall for each threshold. For imbalanced datasets, a high AUROC value can be obtained even if a model does not have a discrimina- tion ability. Unlike AUROC, AUPR is robust to imbalanced datsets.

4.2 CIFAR-10-Corruption

Ovadia et al. [19] have made a benchmark for uncertainty quantification under distributional shifts, and this experiment follows their benchmark. The benchmark employs CIFAR-10-Corruption (CIFAR-10-C) [7] for evaluating the quality of uncertainty based on the ECE. CIFAR-10-C is composed of CIFAR-10, transformed by 16 different types of perturbations whose shift intensity changes up to five levels.

Since the baselines in the benchmark are implemented with ResNet20-v1, the proposed model adopts ResNet20-v1 as an encoder. Additionally, the proposed model employs 6-layer Maksed Autoregressive flow (37,104 parameters) [20]. For the hyperparameters $\boldsymbol{\alpha}_k$ of the conditional base distribution for the class k, a higher value is assigned to the kth dimension of $\boldsymbol{\alpha}_k$:

$$\alpha_{ki} = \begin{cases} 7.0, & \text{if } k = i \\ 0.5, & \text{otherwise} \end{cases} \tag{4.1}$$

For Dropout, Ensembles, and BBB, a pretrained model is employed in the benchmark data. For PostNet, an official implementation code is used to train the model over 200 epochs and the five best models based on the accuracy is selected. As CTNF is composed of an encoder and flows, the encoder is trained for 300 epochs and then

the parameters of the flows are optimized for 3 epochs. For distributionally robust optimization, we set $\gamma = 1.5$, step size $\eta = 0.01$ with 5 iterations. With these five models for each method, the performance of the model is evaluated by three types of corruption: contrast, speckle noise, and Gaussian blur. The result is in the box plots, which represent the accuracy and ECE for CIFAR-10-C, as shown in Figure 4.1

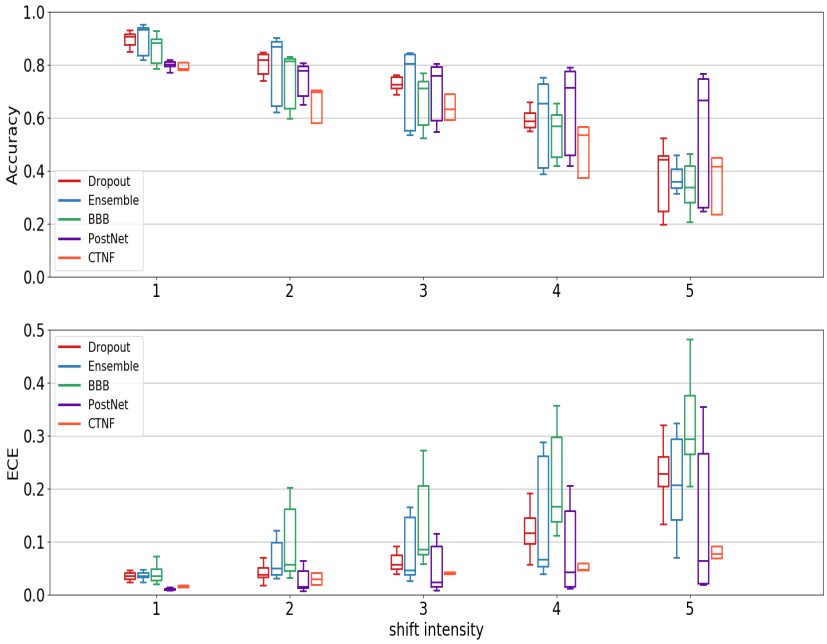

Figure 4.1: Comparison of accuracy and ECE on CIFAR-10. Shift intensity indicates a strength of corruption on CIFAR-10.

Although Dropout, Ensemble, and BBB yield comparable low values of ECE when the shift intensity is mild, their ECE values become larger at the highest shift. A large ECE value at a higher shift indicates that the models yield unreliable predictions under a large distributional shift. Also, the result reports a large change in ECE values of PostNet. This is because the PostNet yields low ECE values on speckle noise and gaussian blur type corruptions, but a high ECE value on contrast corruption. It in-

dicates that the reliability of PostNet's confidence depends on the corruption types. CTNF continues to achieve low ECE values regardless of corruption types and shift intensity. It shows that CTNF provides a consistently well-calibrated classification results. It follows that our model provides a more reliable uncertainty estimates under distributional shifts.

4.3 MNIST Rotation

As in the CIFAR-10 experiment, the benchmark is used for the traditional methods and the official implementation code is employed for PostNet. CTNF adopts LeNet as an encoder [14], which has the same architecture in the benchmark. In addition, a 4-layer Maksed Autoregressive Flow is added. PostNet is trained for 50 epochs and the five best models are selected based on the accuracy. Also, the encoder and flow of the CTNF are trained for 40 epochs and 3 epochs, respectively. We set α_{ki} = 6.0 if $k = i$, and 0.5 if $k \neq i$. Also, $\eta = 0.01$ and $\gamma = 1.4$ are assigned with 5 iterations for DRO. The accuracy and ECE are evaluated while rotating MNIST images by increasing the angle of rotation by 15 degrees.

Figure 4.2 shows a result for MNIST. Dropout, Ensemble and PostNet indicate low ECE values and high accuracy when MNIST images are rotated slightly, but their ECE values increase as we rotate the images. This implies that they tend to yield overconfident predictions at large distributional shifts. CTNF has larger ECE values than other methods when the rotation angle is 15 or 30 degrees; However, the ECE values of CTNF barely change as the rotation angle increases. This implies that CTNF decrease the prediction's confidence as the image rotates and yield reliable confidence at large distributional shifts.

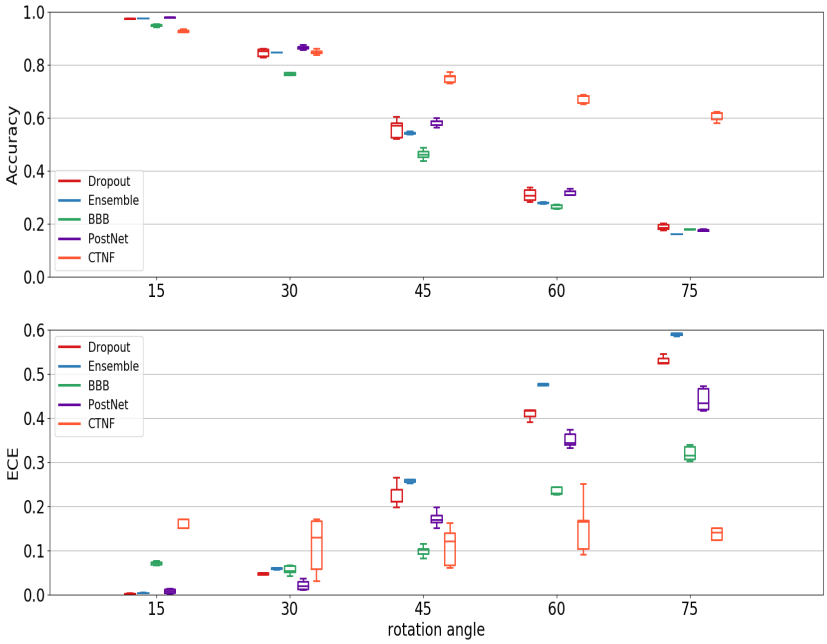

Figure 4.2: Comparison of (a) accuracy and (b) ECE on MNIST: We evaluate the accuracy and ECE while rotating the images.

4.4 Out of distribution detection

First, the baselines and CTNF are trained in the same way as in section 4.2 and 4.3, and these models are tested on various OOD datasets. We evaluate OOD detection performance based on the FPR, AUROC, and AUPR using the threshold-based detectors [15].

Table 4.1 summarizes OOD detection performance for each model. PostNet trained with MNIST outperforms other models for F-MNIST, NotMNIST, and EMNIST. However, its performance degrades when trained with CIFAR-10. Although the proposed method does not perform as well as PostNet on MNIST, OOD detection performance of the CTNF on MNIST is comparable to other methods, and CTNF performs bet-

ID Model	OOD	FPR at 95% TPR ↓	AUROC↑	AUPR Out↑	AUPR In↑
		Dropout / Ensemble / BBB / PostNet / CTNF			
MNIST	F-MNIST	12.0 / 5.90 / 25.3 / **0.09** / 22.1	97.6 / 98.6 / 94.0 / **99.5** / 96.7	97.3 / 98.2 / 94.3 / **99.2** / 96.5	97.3 / 98.8 / 93.7 / **99.6** / 97.0
	NotMNIST	18.2 / 15.0 / 19.7 / **0.23** / 1.9	96.3 / 96.6 / 96.7 / 99.4 / **99.6**	95.9 / 96.0 / 95.7 / 99.1 / **99.6**	94.7 / 96.1 / 97.4 / **99.6** / **99.6**
	EMNIST-letters	27.1 / 26.1 / 77.6 / **1.66** / 21.9	94.3 / 93.4 / 67.1 / **99.1** / 96.1	93.8 / 93.0 / 69.0 / **98.9** / 96.2	90.3 / 89.4 / 61.2 / **99.3** / 96.0
CIFAR-10	SVHN	58.7 / 36.6 / 54.1 / 81.9 / **9.3**	91.1 / 94.9 / 91.8 / 78.3 / **98.1**	86.1 / 92.1 / 93.7 / 72.7 / **98.4**	93.5 / 96.2 / 88.4 / 81.6 / **98.0**
	LSUN	53.5 / 27.0 / 60.0 / 78.2 / **26.6**	90.5 / **96.0** / 87.7 / 80.1 / 94.7	88.1 / 95.0 / 90.0 / 75.3 / **95.1**	92.4 / **96.8** / 85.4 / 82.4 / 95.1
	TinyImageNet	63.9 / 31.7 / 69.0 / 82.9 / **17.1**	87.0 / 95.3 / 82.8 / 76.3 / **96.1**	83.9 / 93.9 / 85.5 / 71.4 / **96.6**	89.5 / **96.3** / 80.1 / 78.8 / 95.9

Table 4.1: Comparison of OOD detection performance for the baselines and CTNF. AUPR-in and AUPR-out indicates that a positive example is specified as in-distribution and out-of-distribution, respectively. The result values presented as percentages.

ter than the baselines for SVHN, LSUN, and TinyImageNet. This implies that CTNF consistently detects unseen OOD data well.

4.5 Ablation study

4.5.1 Effect of the Dirichlet distribution

In this subsection, the effects of the Dirichlet distribution as a base distribution of normalizing flows is investigated. The accuracy and calibration error of our flow model are compared by changing the base distribution of the flows. The two models are the same except for the base distribution and normalization layer, and the DRO method for this experiment is not used because we wish to analyze only the effects of the base distribution. For the Gaussian mixture parameters, we randomly initialize the means sampling from the standard normal distribution, and set the covariance matrices as an identity matrix.

Figure 4.3 shows the accuracy and calibration error of the two models, whose base distributions are different. The results show that the accuracy of the two models differs little and the Dirichlet distribution achieves a lower ECE than the Gaussian mixture. This indicates that the model with the Dirichlet base distribution yields a more reliable uncertainty estimates under distributional shifts.

4.5.2 Effect of distributionally robust optimization

In this subsection, the effect of the DRO for OOD detection is analyzed. OOD detection performance using two models is compared: one with DRO and one without DRO. Both models use the same pretrained encoder, but the flow is trained with or without DRO.

Table 4.2 shows OOD detection performance of the two models. Without DRO, the model cannot detect OOD examples well, as indicated by the high FPR and low AUROC. When applied to DRO, the FPR is reduced and the AUROC is increased

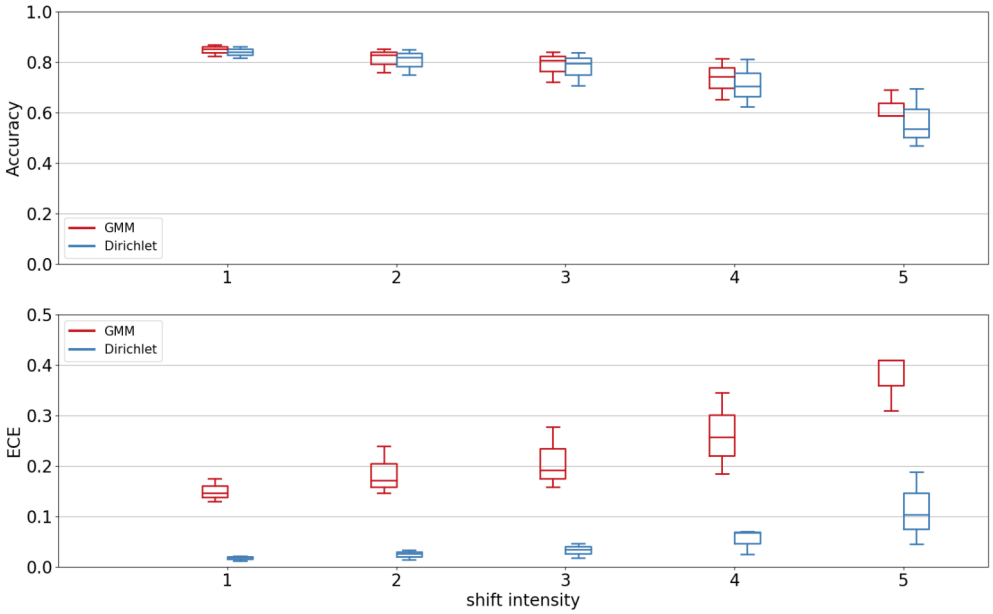

Figure 4.3: Comparison of accuracy and ECE differing by the base distribution of conditional normalizing flows. We use three types of the corruption from CIFAR-10-C: contrast, speckle noise, and gaussian blur.

In	OOD	FPR95 ↓	AUROC ↑
		Base / +DRO	
CIFAR10	SVHN	68.7 / 9.3	85.9 / 98.1
	LSUN	67.0 / 26.6	85.0 / 94.7
	TinyImageNet	71.8 / 17.1	84.7 / 96.1

Table 4.2: Comparison of FPR and AUROC of the proposed models with and without DRO.

significantly. This verifies that the proposed DRO method can improve OOD detection performance of the proposed model.

Chapter 5

Conclusion

In this paper, we presented contrastive normalizing flows (CTNF) which transform the input to the latent points and make the predictions with their confidence in the latent space. The Dirichlet distribution is used as the base distribution of conditional normalizing flow. This increased the expressivity of conditional normalizing flow such that the resulting flow yielded more reliable confidence than the Gaussian mixture based flows. Also, distributionally robust optimization is applied to improve OOD detection performance of CTNF. Through the ablation study, we empirically showed that our distributionally robust optimization scheme could improve OOD detection performance of the proposed model. In addition, the experiments demonstrated that CTNF yielded reliable uncertainty estimates under distributional shifts.

Bibliography

[1] C. BLUNDELL, J. CORNEBISE, K. KAVUKCUOGLU, AND D. WIERSTRA, *Weight uncertainty in neural networks*, in International Conference on Machine Learning, vol. 37, JMLR. org, 2015, pp. 1613–1622.

[2] N. BOUGUILA, D. ZIOU, AND J. VAILLANCOURT, *Novel mixtures based on the dirichlet distribution: Application to data and image classification*, in Machine Learning and Data Mining in Pattern Recognition, P. Perner and A. Rosenfeld, eds., Berlin, Heidelberg, 2003, Springer Berlin Heidelberg, pp. 172–181.

[3] B. CHARPENTIER, D. ZÜGNER, AND S. GÜNNEMANN, *Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts*, arXiv preprint arXiv:2006.09239, (2020).

[4] T. CHEN, S. KORNBLITH, M. NOROUZI, AND G. HINTON, *A simple framework for contrastive learning of visual representations*, arXiv preprint arXiv:2002.05709, (2020).

[5] Y. GAL AND Z. GHAHRAMANI, *Dropout as a Bayesian approximation: representing model uncertainty in deep learning*, in International Conference on Machine Learning, 2016, pp. 1050–1059.

[6] C. GUO, G. PLEISS, Y. SUN, AND K. Q. WEINBERGER, *On calibration of modern neural networks*, arXiv preprint arXiv:1706.04599, (2017).

[7] D. HENDRYCKS AND T. DIETTERICH, *Benchmarking neural network robustness to common corruptions and perturbations*, arXiv preprint arXiv:1903.12261, (2019).

[8] J. M. HERNÁNDEZ-LOBATO AND R. ADAMS, *Probabilistic backpropagation for scalable learning of bayesian neural networks*, in International Conference on Machine Learning, 2015, pp. 1861–1869.

[9] M. D. HOFFMAN, D. M. BLEI, C. WANG, AND J. PAISLEY, *Stochastic variational inference*, Journal of Machine Learning Research, 14 (2013), pp. 1303–1347.

[10] P. IZMAILOV, P. KIRICHENKO, M. FINZI, AND A. G. WILSON, *Semi-supervised learning with normalizing flows*, arXiv preprint arXiv:1912.13025, (2019).

[11] P. KHOSLA, P. TETERWAK, C. WANG, A. SARNA, Y. TIAN, P. ISOLA, A. MASCHINOT, C. LIU, AND D. KRISHNAN, *Supervised contrastive learning*, arXiv preprint arXiv:2004.11362, (2020).

[12] P. KIRICHENKO, P. IZMAILOV, AND A. G. WILSON, *Why normalizing flows fail to detect out-of-distribution data*, 2020.

[13] B. LAKSHMINARAYANAN, A. PRITZEL, AND C. BLUNDELL, *Simple and scalable predictive uncertainty estimation using deep ensembles*, in Advances in Neural Information Processing Systems 30, 2017, pp. 6402–6413.

[14] Y. LECUN, L. BOTTOU, Y. BENGIO, AND P. HAFFNER, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.

[15] S. LIANG, Y. LI, AND R. SRIKANT, *Enhancing the reliability of out-of-distribution image detection in neural networks*, in International Conference on Learning Representations, 2018.

[16] A. MALININ AND M. GALES, *Predictive uncertainty estimation via prior networks*, in Advances in Neural Information Processing Systems, 2018, pp. 7047–7058.

[17] E. NALISNICK, A. MATSUKAWA, Y. W. TEH, D. GORUR, AND B. LAKSHMINARAYANAN, *Do deep generative models know what they don't know?*, in International Conference on Learning Representations, 2019.

[18] D. NIELSEN, P. JAINI, E. HOOGEBOOM, O. WINTHER, AND M. WELLING, *Survae flows: Surjections to bridge the gap between vaes and flows*, in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds., vol. 33, Curran Associates, Inc., 2020, pp. 12685–12696.

[19] Y. OVADIA, E. FERTIG, J. REN, Z. NADO, D. SCULLEY, S. NOWOZIN, J. DILLON, B. LAKSHMINARAYANAN, AND J. SNOEK, *Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift*, in Advances in Neural Information Processing Systems, 2019, pp. 13991–14002.

[20] G. PAPAMAKARIOS, T. PAVLAKOU, AND I. MURRAY, *Masked autoregressive flow for density estimation*, in Advances in Neural Information Processing Systems, 2017, pp. 2338–2347.

[21] A. SINHA, H. NAMKOONG, R. VOLPI, AND J. DUCHI, *Certifying some distributional robustness with principled adversarial training*, arXiv preprint arXiv:1710.10571, (2017).

초 록

불확실성 측정의 목표는 좀 더 신뢰할만한 딥러닝 모델을 설계하기 위함이다. 신뢰도가 높은 모델은 예측을 할 때 예측값의 불확실성을 같이 출력해서 불확실성을 바탕으로 해당 예측값을 믿을지 말지 결정한다. 딥러닝 모델에서 불확실성을 측정하기 위해 여러 연구들이 시행되었지만, 최근 연구들은 데이터 분포가 바뀐 상황에서 전통적인 방법들이 측정한 불확실성의 신뢰도가 부족하다는 것을 실험적으로 보였다. 본 논문에서는 데이터 분포가 변한 상황에서 신뢰도가 높은 불확실성을 측정하는 Contrastive Normalizing Flow 모델을 제시한다. 제시한 모델은 잠재 공간에서 불확실성을 측정한다. Contrastive learning으로 학습된 인코더는 이미지를 잠재 공간으로 대응시키고, 생성 분류 모델은 예측 분포를 normalizing flow를 사용해서 모델링한다. 그리고 out-of-distribution을 검출하기 위해 distributionally robust optimization 방법을 모델에 적용한다. 실험에서는 두 가지 분포가 변하는 상황(covariate shift, out-of-distribution)을 고려하고, 해당 상황에서 제시한 모델의 견고함을 실험적으로 증명한다.

주요어: 강건성, 불확실성, 상관 없는 데이터

학번: 2019-29928