Notation

The training dataset $\mathcal{D} = \{x_n, y_n\}_{n=1}^N$

$x \in R^D$

For classification problem, $y \in \{1, \dots, K\}$. For regression problem, $y \in R$.

Given the input x, $p_\theta(y|x)$ defines the predictive distribution over the label.

Basic Concepts

1. Calibration refers to the difference between the model forecasts and empirical long-run frequencies.

2. Well-calibration is a principle we aim to have in our model. In practice, calibration is measured by scoring rule.

Proper Scoring Rule

1. Scoring rule is a function $S(p_\theta, (y, x)) \in R$ that measures the quality of the predictive distribution $p_\theta(y|x)$ relative to an event $y|x \sim q(y|x)$ where $q(y, x)$ is the true distribution on the tuple (y, x).

2. By convention, a high numerical scoring rule is better.

3. Overloading notation, the expected scoring rule is $S(p_\theta, q) = \int q(y, x) S(p_\theta, (y, x)) dxdy/$

4. A proper scoring rule is one where $S(p_\theta, q) \leq S(q, q)$ with equality iff $p_\theta(y|x) == q(y|x), \forall x, y$.

5. $S(p_\theta, q) = \log p_\theta(y|x)$ is a proper scoring rule because of Gibbs inequality

$$\int q(y, x) \log p_\theta(y|x) dxdy = \int q(x) q(y|x) \log p_\theta(y|x) dxdy = E_{x \sim p(x)}[\int q(y|x) \log p_\theta(y|x) dy] \leq E_{x \sim p(x)}[\int q(y|x) \log q(y|x) dy]$$

6. In multi-class K-way classification, $S(p_\theta, (y, x)) = -\frac{1}{K} \sum_{k=1}^K (\delta_{k=y} - p_\theta(y = k|x))^2$ is a proper scoring rule, referred to as the Brier score.

Training criterion for regression

1. Given an input x, their network predict a mean $\mu(x)$ and variance $\sigma^2(x)$.

2. They update θ to maximize the log-likelihood $\log p_\theta(y|x)$

Adversarial Training

1. Fast gradient sign method for generating adversarial example: Given $x, y, l(\theta, x, y)$, the adversarial example is generated as $x' = x + \epsilon sign(\boldsymbol{\nabla}_x l(\theta, x, y))$

2. Adversarial training can be interpreted as a computationally efficient solution to smooth the predictive distribution by increasingly the log likelihood of the target around an $\epsilon-$neigborhood of the observed training samples.

Ensemble Training and Prediction

1. Generally, there are 2 classes of ensemble methods: randomization-based where ensemble members can be trained without any interaction and boosting-based approach where the ensemble members are fit sequentially.

2. The paper uses randomization-based approach because it is easier to parallelize.

3. They want a randomization-scheme that de-correlates the predictors of the ensemble members but ensure that each ensemble members have high accuracy.

4. A popular randomization-scheme is bagging (aka bootstrapping), where ensemble members are trained on different subset of the training set.

5. Bagging degrades the performance of ensemble members. If the subset is selected by sampling N times uniformly with replacement from a dataset with N items, then the number of unique data points in the subset is $0.632 \times N$ on average.

6. They found that having random initialization and random minibatch sampling were enough to ensure that the ensemble members are sufficiently diversed.

7. They treat the ensemble as a uniformly weighted mixture model and combine the predictions as $M^{-1} \sum_m p_{\theta_m}(y|x, \theta_m)$. For classification, this corresponds to averaging the predicted probabilities. For regression, the prediction is a mixture of Gaussian. They further approximate the ensemble prediction as a Gaussian whose mean and variance are given by the means and variances of the ensemble members.

Experiments

1. When performing adversarial training, they do not set ϵ to a fixed number across the input dimensions. They mention this is problematic if the input dimensions have different range. For each dimension, they thus choose ϵ to be the range of the dimension multiplied by a small constant.

2. On input from unknown classes, they use the entropy of the predictive distribution to measure how well the model detects out-of-distribution samples. For unseen classes, they expect the predictive distribution to be closer to uniform compared to seen classes.

3. For known classes, both ensemble method and MC-dropout have low entropy, as expected.

4. For unknown classes, as M increases, the entropy of the deep ensembles increases much faster than MC-dropout, indicating that the ensemble method is better suited to handle test examples from unseen classes.

5. MC-dropout produces high confidence prediction for some of the test examples from unseen classes.

6. For test examples from unseen classes, they observe that as the size of the ensembles increases, the entropy of the predictive distribution increases and the maximum predicted probability decreases.

7. To evaluate the usefulness of the predictive uncertainty for decision making, they consider a setting where the model's accuracy is only evaluated on test examples where the model's confidence in its prediction is above a user-specified threshold.

8. In this setting, given a pre-specified confidence threshold, they show that the ensemble approach outperforms MC-dropout in terms of prediction accuracy.

Comment

1. It is unclear why this method outperforms MC-dropout.

2. In figure 6 (accuracy vs confidence), there is still a large room for improvement since at 0.9 confidence threshold, the best ensemble model still has accuracy less than 90% on MNIST.