DeepBayes Summer School - Paper Assignment

Lucas P. Cinelli

April 17, 2019

Paper of choice: **Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles**, from Balaji Lakshminarayanan, Alexander Pritzel, Charles Blundell.

1 Question 1

How do authors change the NN to make it capable to estimate uncertainty for regression tasks? What is the distribution on the outputs, as defined by the NN architecture and loss? What distribution on the outputs would be induced by an ensemble of such NNs?

Instead of employing a Bayesian approach to estimate uncertainty, the authors rely on a frequentist view of the problem and use ensembles to obtain a predictive uncertainty measure. Besides, the authors adapt each neural network of the ensemble to output two values, the predicted mean μ and variance σ^2, and treat the observed value $\mathbf{x}$ corresponding to the model's input as a sample from a heteroscedastic Gaussian (non-equal noise level across the data). Hence, the log-likelihood function used to optimize the model becomes

$$\log p(y_n|\mathbf{x}_n) = \log \mathcal{N}(y_n; \mu(\mathbf{x}_n), \sigma^2(\mathbf{x}_n)) = -\frac{\log \sigma^2(\mathbf{x}_n)}{2} - \frac{(y - \mu(\mathbf{x}_n))^2}{2\sigma^2(\mathbf{x}_n)} + \text{const.}. \tag{1}$$

From the above formula, we can view the role of $\sigma^2(\mathbf{x})$ as diminishing the cost associated with misprediction, so if the model is unsure about its prediction, increasing $\sigma^2(\mathbf{x})$ would lower the toll. Still, increasing the value of $\sigma^2(\mathbf{x})$ does not come for free, the first RHS also imposes a cost on large $\sigma^2(\mathbf{x})$, otherwise setting it to be arbitrarily large would be beneficial.

Furthermore, they treat the ensemble as a uniformly-weighted mixture model and combine the predictions of each model. The resulting distribution, which corresponds to a mixture of Gaussian in regression tasks, is then approximated by a Gaussian with mean and variance given by the mean and variance of the ensemble. Thus, in the above formula we should actually use the mean and variance of the ensemble instead of that of the individual model as we show here.

2 Question 2

What are adversarial examples? What is the purpose of using them to train the ensemble? Can an object with an unchanged prediction be an adversarial example?

Adversarial examples are samples that look similar to the original training examples but are misclassified by the model. Such examples are usually forced by altering the data samples in a specific manner, as for example the Fast Gradient Sign method proposed by Goodfellow, which, intuitively, creates a new training example by adding a perturbation along a direction which the network is likely to increase the loss.

The authors alter the optimization function to include a new term corresponding to the log-likelihood of the correct label given the adversarially modified sample, that is

$$\alpha \log p(y_n|\mathbf{x}_n) + (1-\alpha) \log p(y_n|\mathbf{x}_n + \Delta\mathbf{x}_n), \tag{2}$$

for a $(\mathbf{x}_n, y_n)$ data point pair, where $\Delta\mathbf{x}_n$ is the adversarial perturbation inflicted to $\mathbf{x}_n$, i.e., the modification computed by the Fast Gradient Sign method.

This new optimization function trades-off between the usual maximum likelihood score and an adversarial score, inducing the log-likelihood of y to be high even in unfavourable cases. This has the effect of smoothing the prediction distribution along the most critical direction, i.e., where the loss is higher. This is helpful even if the prediction, $\mu(\mathbf{x})$, remains unchanged, since $\sigma^2(\mathbf{x})$ will probably change given that $\mathbf{x}_n + \Delta\mathbf{x}_n$ represents a more challenging example to the model, causing it to be more uncertain. Hence, the optimization function will change even if the prediction does not.

3 Question 3

Let's imagine that somebody collected a dataset with many out-of-domain images or images with wrong labels. How can the proposed uncertainty estimation method be applied to clean the dataset from such objects?

If the data set contains out-of-domain images or images with wrong labels, their distribution are very different from that of the rest of the data set, thus the trained model should be fairly uncertain about its prediction, that is, the different elements of the ensemble should not agree among themselves and give different predictions. Thus, we could for example measure the predictive entropy of the samples for all data set. We expect to see low entropy when the samples are in-distribution, which means the elements of the ensemble agree, while high entropies when they disagree, meaning that our model is uncertain and that sample probably is out-of-distribution (or has the wrong label).