Deep|Bayes 2019

Practical Assignment

Shubham Gupta

shubhamgblr@gmail.com

Disclaimer:

For this assignment, I have used the following resources to help aid my understanding of the paper and the concepts present in it:

- The orignal paper present here.
- The poster session organized by the authors at NIPS 2017 present here
- The NIPS 2017 conference video present here(start from the 01:00:00 mark)
- A quick summary of the paper present here
- Instacart Tech blog present here
- A review of the paper by Alberto Arrigoni present here

Q1

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?

Answer

To estimate uncertainity for regression tasks, the authors propose a change in the final layes of the NN in which two values are given from the final layes, the prediction mean $\mu(x)$ the predicted variance $\sigma_\theta^2(x)$ as Mean Squared Error(MSE) itself will not be able to capture the predictive uncertainity.

The distribution of the outputs changes from a single value to a Gaussian distribution with shape parameter $\mu(x)$ and scale parameter $\sigma_\theta^2(x)$. Due to this, we will compute the negative log likelihood as follows:

$$-logp_\theta(y_n|x_n) = \frac{log\sigma_\theta^2}{2} + \frac{y - \mu_\theta(x)}{2\sigma_\theta^2(x)} + constant$$

For an ensemble of such NN's, we will get a mixture of Gaussian distributions. Further, the author's apprxomiate this mixture as a single gaussian distribution with the following mean and variance:

$$\mu_*(x) = M^{-1} \sum_m \mu_{\theta_m}(x)$$

$$\sigma_*^2(x) = M^{-1} \sum_m (\sigma_{\theta_m}^2(x) + \mu_{\theta_m}^2(x)) - \mu_{\theta_*}^2(x)$$

Q2

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?

Answer

Advesarial samples are data points that are close to training examples but misclassified by NN. In some cases, they could also be optical illusions that are designed by attackers to folder NN systems.

Advesarial samples can be used to augment training samples with new data. This is useful because

- It can increase the number of datapoints available
- It helps improve classifier robustness.
- It helps obtain smoother predictive distributions. This is because advesarial examples are generated by using the fast gradient sign method with parameter ϵ. These samples(termed x') can be treated as additional training samples with the label y i.e (x', y). These samples will help in increasing the likelihood of the target of these samples thereby leading to smoother predictive distributions.

Yes, an object with an unchanged prediction can be an adversarial example. As part of the fast gradient sign method, an andesarial example is generated by:

$$x' = x + \epsilon sign(\nabla l(\theta, x, y))$$

If the value of ϵ is very small, then the advesarial sample will be very much like the training sample(to the human eye), and the prediction for this sample will also be the same as that of the training sample.

Q3

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?

Answer

Assuming we have an ensemble of NN's ready which have been trained on a dataset containing data with the correct labels. We can then use this method

to correct the labels as follows:

- Pick a threhold value say t. This threshold value will the he minimum possible predictive uncertainity we are comfortable accepting. When the prediction by the NN's for an image from the dataset is greater than the value t, then we can safely assume that the prediction is the true label for the image
- How to pick the threshold value?
    - This could again be modelled as a sampling process.
    - Sample images from the dataset. Pick a random number of threshold values t.
    - Run the ensemble of NN's on this sample for all the t values.
    - Repeat this process with multiple samples of the dataset. We could use grid search to figure out the possible threshold values.
    - Once we have the results, for each of the sample dataset, we could manually check which threshold value classifies the most number of images correctly(similar to a manual accuracy and recall check).
