Abstract

This supplementary file contains the pseudo-code for our learnable Bernoulli dropout (LBD), run-time comparison with other existing dropout models, as well as additional experimental results. The code for the experiments can be found at the following link.

1 Pseudo-code of LBD training

Algorithm 1 outlines the major steps for learning dropout rates and other network parameters in our LBD framework.

Result: \( \theta \) that minimizes \( \mathbb{E}_z \left[ \mathcal{L}(\theta, z | D) \right] \)
where \( \theta = \{ \psi, \alpha \} \) consists of dropout mask parameters \( \alpha \), and all other parameters \( \psi \);
Initialize \( \theta \) randomly;
while not converged do
    Sample a mini-batch of size \( M \) from the data;
    Sample \( u_{ijk} \sim \text{Unif}[0,1] \) for \( i = 1,\ldots,M \),
    \( j = 1,\ldots,L \), \( k = 1,\ldots,K_{j-1} \);
    \( g_\psi = \nabla_\psi \mathcal{L}(\theta, 1_{u < \sigma(\alpha)}) \);
    \( g_\alpha = \left( \mathcal{L}(\theta, 1_{u > \sigma(-\alpha)}) - \mathcal{L}(\theta, 1_{u < \sigma(\alpha)}) \right) \left( u - 1/2 \right) \);
    \( \psi = \psi - r_\psi g_\psi \);
    \( \alpha = \alpha - r_\alpha g_\alpha \);
end

Algorithm 1: Parameter estimation in Learnable Bernoulli Dropout scheme.

2 Additional toy example results

The bias and mean squared error of the estimates for \( \alpha_2 \) from LBD and Concrete calculated by 200 Monte Carlo samples for the toy example in Section 3.1 are shown in Figure 1.
3 Additional semantic segmentation results

Figure 2 provides the visualization of the example outputs from FC-DenseNet-103 with different dropout schemes for semantic segmentation on the CamVid dataset (Section 3.1.2). The bottom panels depict the segmentation results while the top panels visualize the uncertainty (predictive entropy).

4 Run-time comparison

Here, we discuss how much run-time change we observed empirically when using LBD compared with other dropout variants.

For image classification using VGG19 on the CIFAR-10 dataset (Section 3.2.1), all dropout variants had almost the same training run time.

For the semantic segmentation with FC-DenseNet-103 on CamVid dataset (Section 3.2.2), LBD, Concrete and Gaussian dropouts had the same training run time, with a 4.5% increase in run time compared with the regular dropout.

In the collaborative filtering experiments (Section 3.3), the SIVAE with LBD (SIVAE+LBDrop), SIVAE with Concrete dropout (SIVAE+CDrop), and SIVAE with Gaussian dropout (SIVAE+GDrop) all had the same training run time.

5 Regression on UCI Datasets

In addition to the reported results in the tasks described in the main text, we also have tested different dropout schemes for regression on various UCI datasets [1]. In this setting, following [2] we implemented our LBD on a fully connected network with 2 hidden layers, with 50 hidden units each. We compared our method with other dropout methods, such as Concrete dropout, Gaussian dropout, and MC dropout, on the same architecture. We randomly split each dataset into 90% training and 10% testing data for 20 times (5 times for Naval Propulsion and Protein Structure). In each split, we train the fully connected architecture using the different dropout methods for 65 epochs using a single sample for each batch. The RMSE of each method is shown in Table 1. As we can see, all the methods that optimize the dropout parameters perform very similarly. The improvement of one method over the others does not appear to be statistically significant.

Table 1: Root Mean Square Error (RMSE) of different methods trained on various UCI datasets. Standard deviations of the RMSE are shown in parentheses

<table>
<thead>
<tr>
<th>Dataset</th>
<th>N</th>
<th>d</th>
<th>Average RMSE (Standard Deviation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston Housing</td>
<td>506</td>
<td>13</td>
<td>4.26 (1.21) 3.55 (1.06) 3.60 (1.27)</td>
</tr>
<tr>
<td>Concrete Compression Strength</td>
<td>1,030</td>
<td>8</td>
<td>7.47 (0.71) 4.91 (0.63) 4.94 (0.51)</td>
</tr>
<tr>
<td>Energy Efficiency</td>
<td>768</td>
<td>8</td>
<td>2.24 (0.21) <strong>0.87 (0.13)</strong> 1.01 (0.15)</td>
</tr>
<tr>
<td>Naval Propulsion</td>
<td>11,934</td>
<td>16</td>
<td><strong>0.01 (0.00)</strong> <strong>0.01 (0.00)</strong> <strong>0.01 (0.00)</strong></td>
</tr>
<tr>
<td>Combined Cycle Power Plant</td>
<td>9,568</td>
<td>4</td>
<td>4.79 (0.22) 4.06 (0.16) <strong>4.05 (0.19)</strong></td>
</tr>
<tr>
<td>Protein Structure</td>
<td>45,730</td>
<td>9</td>
<td>4.83 (0.05) 4.09 (0.06) 4.06 (0.04)</td>
</tr>
<tr>
<td>Wine Quality Red</td>
<td>1,599</td>
<td>11</td>
<td><strong>0.63 (0.04)</strong> 0.67 (0.04) <strong>0.63 (0.05)</strong></td>
</tr>
<tr>
<td>Yacht Hydrodynamics</td>
<td>308</td>
<td>6</td>
<td>4.87 (1.32) 1.99 (0.68) 1.78 (1.11)</td>
</tr>
</tbody>
</table>

Concrete dropout (SIVAE+CDrop), and SIVAE with Gaussian dropout (SIVAE+GDrop) all had the same training run time.
References
