Dropout as variational inference

Apostolos Psaros

1 Forward pass with dropout

Consider a NN with a single hidden layer. The hidden layer weights and biases are $\boldsymbol{m}_1$ and $\boldsymbol{b}$, respectively. These are vectors of size K, the width of the layer. The output weights are $\boldsymbol{m}_2$, also of size K. For doing a forward pass with dropout we sample a vector $\hat{\boldsymbol{\epsilon}} \sim p(\boldsymbol{\epsilon})$ of dimension K. The elements of $\hat{\boldsymbol{\epsilon}}$ take value 0 with probability $0 \leq p \leq 1$. Given the output of the hidden layer $\boldsymbol{h} = \sigma(x\boldsymbol{m}_1 + \boldsymbol{b})$, we set a proportion of it to zero, i.e., $\hat{\boldsymbol{h}} = \boldsymbol{h} \odot \hat{\boldsymbol{\epsilon}}$. Finally, $f_{\mathcal{H}}(x_i; \boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}) = \boldsymbol{m}_2^T \hat{\boldsymbol{h}}$, where $\boldsymbol{\theta} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{b}\}$.

2 Training with dropout

Training involves obtaining $\boldsymbol{\theta}$ by minimizing

$$\mathcal{L}_{dropout}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{1}{2} |y_i - f_{\mathcal{H}}(x_i; \boldsymbol{\theta}, \boldsymbol{\epsilon})|^2 + Reg(\boldsymbol{\theta}) \tag{1}$$

where $\boldsymbol{\epsilon}$ is a random variable as described above and $Reg(\boldsymbol{\theta})$ is a regularization term. In each iteration, for obtaining the gradient of $\mathcal{L}_{dropout}$ we use a sample $\hat{\mathcal{L}}_{dropout}$ given as

$$\hat{\mathcal{L}}_{dropout}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{1}{2} |y_i - f_{\mathcal{H}}(x_i; \boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}})|^2 + Reg(\boldsymbol{\theta}) \tag{2}$$

3 Equivalence with variational inference

Note that

$$\begin{aligned} f_{\mathcal{H}}(x_i; \boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}) &= \boldsymbol{m}_2^T \hat{\boldsymbol{h}} \\ &= \boldsymbol{m}_2^T diag(\hat{\boldsymbol{\epsilon}}) \boldsymbol{h} \\ &= \hat{\boldsymbol{w}}_2^T \boldsymbol{h} \\ &= f_{\mathcal{H}}(x_i; \hat{\boldsymbol{w}}) \end{aligned} \tag{3}$$

where $\hat{\boldsymbol{w}} = \{\boldsymbol{m}_1, \hat{\boldsymbol{w}}_2, \boldsymbol{b}\}$ is a sample of the random variable $\boldsymbol{w} = \{\boldsymbol{m}_1, \boldsymbol{w}_2, \boldsymbol{b}\}$, with $\boldsymbol{w}_2 = \boldsymbol{m}_2 \odot \boldsymbol{\epsilon}$ and $\hat{\boldsymbol{w}}_2 = \boldsymbol{m}_2 \odot \hat{\boldsymbol{\epsilon}}$. We can therefore transfer the uncertainty from the feature space to the model weights. Therefore, the training loss in each iteration is

$$\hat{\mathcal{L}}_{dropout}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{1}{2} |y_i - f_{\mathcal{H}}(x_i; \hat{\boldsymbol{w}})|^2 + Reg(\boldsymbol{\theta}) \tag{4}$$

which can also be written as

$$\hat{\mathcal{L}}_{dropout}(\boldsymbol{\theta}) = -\beta \sum_{i=1}^{N} \log p(y_i|\hat{\boldsymbol{w}}, x_i, \mathcal{H}) + Reg(\boldsymbol{\theta}) \tag{5}$$

If we write $\boldsymbol{w} = \boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{\epsilon}) = \{\boldsymbol{m}_1, \boldsymbol{m}_2 \odot \boldsymbol{\epsilon}, \boldsymbol{b}\}$, Eq. (5) becomes

$$\hat{\mathcal{L}}_{dropout}(\boldsymbol{\theta}) = -\beta \sum_{i=1}^{N} \log p(y_i|\boldsymbol{g}(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}), x_i, \mathcal{H}) + Reg(\boldsymbol{\theta}) \tag{6}$$

Recall that the loss in Bayes by backprop is

$$\hat{\mathcal{L}}_{BBB}(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \log p(y_i|\boldsymbol{g}(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}_i), x_i, \mathcal{H}) + KL\left(q_{\boldsymbol{\theta}}(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{H})\right) \tag{7}$$

4 Summary

Training with dropout is equivalent to Bayes by backprop

1. with a difference in scale in the summation term
2. with reparametrization $\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{\epsilon}) = \{\boldsymbol{m}_1, \boldsymbol{m}_2 \odot \boldsymbol{\epsilon}, \boldsymbol{b}\}$
3. with prior $p(\boldsymbol{w}|\mathcal{H})$ and approximating distribution $q_{\boldsymbol{\theta}}(\boldsymbol{w})$ such that $KL\left(q_{\boldsymbol{\theta}}(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{H})\right) = Reg(\boldsymbol{\theta})$.

Two more notes:

1. Other stochastic regularization techniques can be recovered with different reparametrizations and $\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{\epsilon})$
2. after training with dropout the NN can be used exactly as a BNN (MC dropout).

Overall, optimizing any NN with dropout is equivalent to *a form* of variational inference.